Flowcode - Mode d'emploi

Sommaire

I) Introduction

I-1) Introduction

Bienvenue dans Flowcode. Flowcode est un logiciel qui vous permet de créer des programmes relativement complexes pour les microcontrôleurs de la famille des PICmicro® d'Arizona Microchip.

Ce fichier d'aide contient les informations dont vous aurez besoin pour utiliser Flowcode afin de programmer des composants PICmicro. Si vous sentez que des choses vous échappent ou si vous avez des questions techniques, merci de contacter multipower@wanadoo.fr.

Comment utiliser ce fichier d'aide
Ce fichier d'aide devra être utilisé en conjonction avec la vingtaine de tutoriaux que nous avons écrits pour faciliter l'apprentissage de Flowcode. Ces tutoriaux sont livrés avec Flowcode et doivent se trouver dans le sous-répertoire TUTORIALS lié au dossier d'installation. Nous vous suggérons de parcourir ces tutoriaux. Travailler successivement chaque tutorial et référez-vous au fichier d'aide pour plus d'informations sur les actions rencontrées dans chacun d'eux. Si vous ne comprenez pas tout immédiatement, pas de panique ! Continuez votre progression et revenez à ce tutorial plus tard.

Aide contextuelle
Dans chaque fenêtre de Flowcode, vous remarquerez de petits boutons repérables par un '?'. Ils vous apporteront une aide contextuelle sur les fonctions concernées.
Vous pouvez aussi à n'importe quel moment presser la touche de fonction F1. L'aide contextuelle correspondant à l'élément actuellement sélectionné sera affichée.

Connaissances préalables
Flowcode est destiné aux utilisateurs qui possèdent les bases suivantes :

Les principes de base de la logique numérique
- Qu'est-ce qu'un PICmicro ?
- A quoi sert un PICmicro et que peut-on faire avec ?
- Un PICmicro a besoin d'une horloge en entrée.
- Un PICmicro possède une circuiterie interne tels que des timers, qui peut être utilisée dans des programmes.
- Les algorigrammes et leur fonctionnement
- Les principes élémentaires de Windows™ comme copier, coller etc.
- Les circuits électroniques simples comme les LEDs, les Interrupteurs, les transistors, etc.

Si vous avez quelques lacunes, pas de panique ! Gardez seulement à l'esprit que peut-être vous ne comprendrez pas tout du premier coup.

I-2) Nouveautés de la version 2

Nous listons ici les principales évolutions de la version 2.

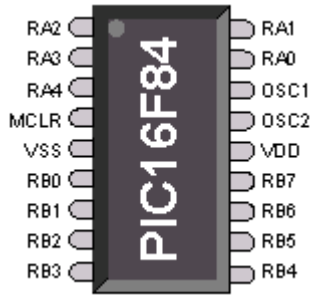
- Nouveaux menus et barres d'icônes pour faciliter la compilation et la configuration.
- Nouveaux PICmicros – la liste des PICmicros cibles a été étoffée.
- Prise en compte de PPPv3 qui supporte nos nouvelles cartes de développement PICmicros sur bus USB.
- Support des tableaux.
- Mise à jour de l'afficheur LCD qui peut à présent afficher des chaînes de caractères et des nombres ainsi que des digits.
- Import des macros amélioré afin de permettre à l'utilisateur de mieux résoudre les conflits de variables.

I-3) Les microcontrôleurs PIC

PIC est la contraction de Programmable Integrated Circuit. Plus généralement, un PICmicro est un composant simple contenant un microprocesseur, de la mémoire RAM (Random Access Memory), de la mémoire ROM (Read Only Memory) et des circuits d'entrées/sorties. Ces composants nécessitent d'être programmés en code hexadécimal (HEX).

Flowcode produit le code hexadécimal nécessaire au PICmicro grâce à un traitement appelé 'Compilation'. Pour compiler et produire le code hexa, Flowcode fait appel à un certain nombre de programmes – un compilateur C et un assembleur. Tout d'abord Flowcode traduit votre algorigramme en langage C, puis en assembleur et ensuite en hexadécimal. Ne vous inquiétez pas si vous n'avez jamais utilisé des langages de programmation comme le C ou l'assembleur – la spécificité de Flowcode, c'est que vous n'avez pas à connaître l'un ou l'autre de ces langages !

Avant de commencer avec Flowcode, vous devez savoir un peu ce qu'est un PICmicro et ce qu'il est capable de faire. Il existe des centaines de types de PICmicro allant du PICmicro très simple à 8 pattes jusqu'à des PICmicros très complexes de 40 pattes. En abordant en premier la section des timers, nous aimons faire référence au PIC16F84. C'est un PICmicro assez simple de 18 pattes comportant 13 pattes d'entrées/sorties qui peuvent être configurées chacune soit comme une entrée, soit comme une sortie. Voyez le schéma ci-dessous :



Les pattes (pins) RB0 à RB7 sont collectivement nommées 'Port B'. Chaque patte peut être configurée en entrée ou en sortie. Quand vous choisissez un PICmicro dans Flowcode, chaque patte sera déclarée soit en entrée soit en sortie pour vous. Le Port B dispose de 8 pattes qu'on peut imaginer comme la représentation d'un octet. Le Port B est en fait contrôlé par un octet dans la mémoire RAM du PICmicro. Pour chaque patte du port, on parle aussi souvent de

«bit » de port, puisque c'est ce moyen qui est utilisé pour la contrôler dans un programme. Par exemple, le bit 4 du port B fait référence à la patte RB3.

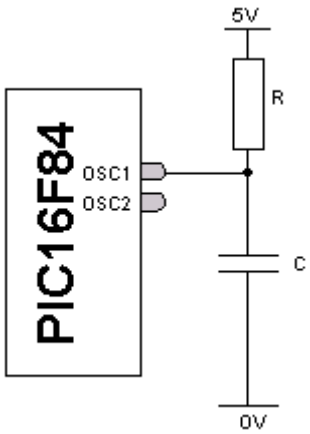
Grâce à Flowcode, vous pouvez modifier l'état de n'importe quel bit à un moment donné. Vous pouvez tout aussi bien modifier l'état de toutes les pattes du Port B en une seule opération en envoyant un nombre compris entre 0 et 255 au Port B (ce qui revient à accéder aux 8 bits de Port B). Pour écrire dans tous les bits du Port B de façon simultanée, vous devez comprendre le mécanisme des nombres binaires. Dans le système binaire, le nombre 0 se représente par le bit le moins significatif d'un octet (ou byte en anglais), le nombre 2 est représenté par le bit 1, le nombre 4 correspond au bit 2 etc. Par exemple, envoyer le nombre 4 au Port B revient à mettre le bit 2 à 1 ; envoyer le nombre 8 revient à mettre le bit 3 à 1 ; envoyer la valeur 6 revient à mettre les bits 1 et 2 à 1.

Les pattes RA0 à RA4 sont collectivement nommées 'Port A'. Le Port A est traité de la même façon que le Port B et les opérations que nous venons de voir sur les bits s'appliquent tout aussi bien au port A. Cependant, le Port A ne dispose que de 5 pattes – bits 0 à 4. Le nombre est limité par le nombre de pattes de ce composant à 18 pattes.

Pattes VSS et VDD : il s'agit des pattes d'alimentation : le positif de la tension d'alimentation (normalement 5V) est appliqué à la patte VDD du boîtier et la masse à la patte VSS.

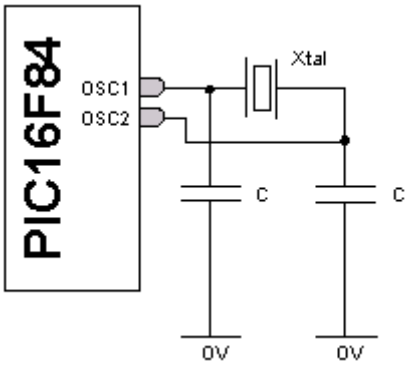
MCLR est la patte de reset. Mettre cette patte à 0V pour reseter le composant – ou pour redémarrer votre programme.

OSC1 et OSC2 sont deux pattes d'horloge. Tous les microcontrôleurs PIC nécessitent une circuiterie de génération d'horloge pour faire tourner leur programme. Il existe plusieurs types d'horloge. Si vous avez besoin d'une solution peu onéreuse, le mieux est d'utiliser un couple résistance - capacité comme montré ci-dessous :



La vitesse d'horloge sera fonction de R, C et de la tension d'alimentation. OSC2 est la patte de sortie d'horloge.

Si vous avez besoin d'une horloge plus précise et tournant plus vite, utilisez une horloge à quartz. Dans ce cas, votre circuiterie ressemblera à ceci :



Dans les deux cas, vous avez besoin de spécifier la fréquence d'horloge et le type d'horloge à la fois dans Flowcode et dans votre programmateur de PIC : la circuiterie interne de chaque système est différente et doit être configurée lors de la programmation du composant.

Le PIC16F84 dispose d'une architecture interne relativement simple. D'autres composants intègrent des ressources internes permettant de concevoir des applications beaucoup plus complexes. Il s'agit par exemples des ressources suivantes :

Convertisseur analogique/digital (A/D) : ce convertisseur analogique/digital 8, 10 ou 12 bits permet d'utiliser des détecteurs externes comme des capteurs de lumière ou de température, qui délivrent en sortie un signal analogique représentant la quantité mesurée.

Interface de communication série plus communément appelée USART (Universal Synchronous/Asynchronous Receiver/Transmitters) c'est à dire un émetteur/récepteur universel de données synchrones ou asynchrones. Cette interface est utilisée pour permettre la connexion de l'application avec la majorité des équipements informatiques classiques.

Reportez-vous à la note technique du PIC16F84 pour plus de détails. Vous pouvez vous procurer les notes techniques à l'adresse **www.microchip.com**.

I-4) Support technique

Le support technique concernant Flowcode peut être obtenu par courrier électronique à : multipower@wanadoo.fr.

Problèmes répertoriés

Compiler avec des droits utilisateur restreints
Si vous utilisez Flowcode avec des droits utilisateur limités ou restreints vous devrez contacter votre administrateur système pour obtenir des privilèges d'accès en lecture/écriture à la clé de registre HKEY_LOCAL_MACHINE\Software\Licenses.
Ce problème concerne les utilisateurs de Windows 2000 et Windows XP.

Il arrive que la liste des derniers fichiers utilisés ne fonctionne pas.
Lorsque vous choisissez un fichier depuis la liste de lancement de Flowcode ou la liste placée en bas du menu 'Fichier' , il se peut que la compilation ne fonctionne pas correctement.
Dans ce cas, ouvrez le fichier à partir de la commande 'Ouvrir' du menu 'Fichier'.
Ce problème semble lié aux droits d'accès de l'utilisateur.

II) Pour commencer

II-1) Concevoir un algorigramme pour un composant PIC

Flowcode vous permet de créer des applications pour des microcontrôleurs en sélectionnant et plaçant des icônes pour créer des programmes simples. Ces programmes peuvent contrôler des périphériques externes connectés au microcontrôleur comme des LEDs, un afficheur LCD etc.

Une fois que l'algorigramme est terminé, Flowcode vous permet de simuler son comportement avant de le compiler, de l'assembler et de le transférer dans un microcontrôleur PICmicro® .

Pour atteindre cet objectif avec Flowcode, il suffit de réaliser les étapes suivantes :

1. Créer un nouvel algorigramme, spécifier le microcontrôleur cible que vous utiliserez.

2. Sélectionner et faire glisser les icônes de la barre d'outils vers l'algorigramme représentatif de votre application.

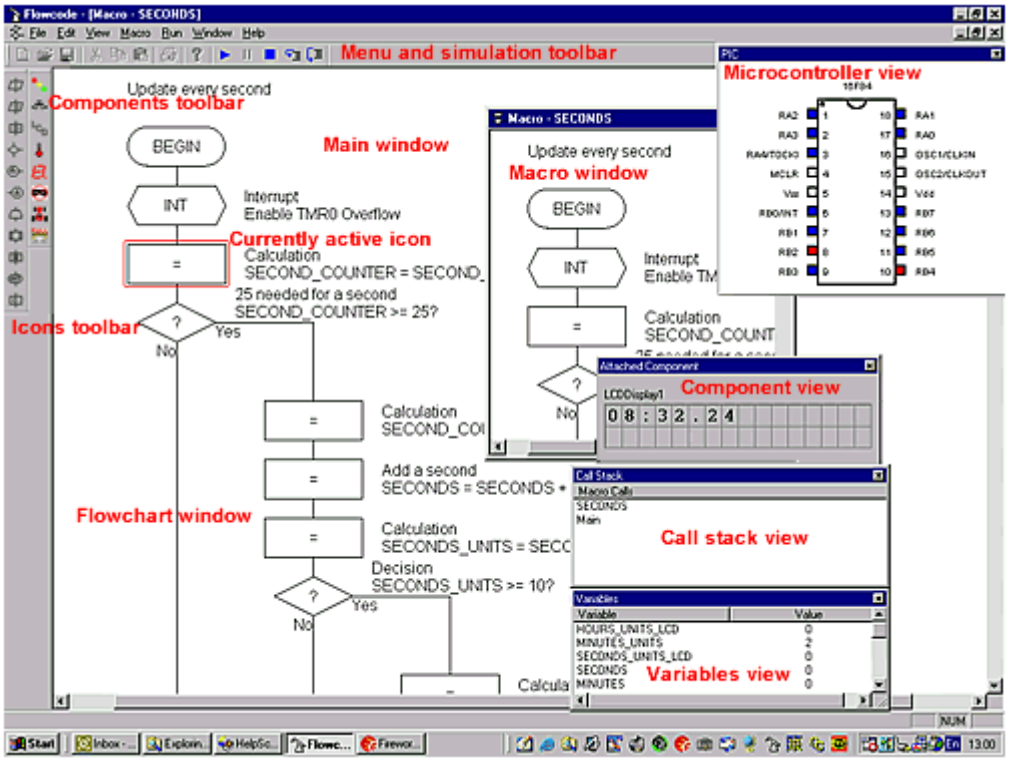
3. Ajouter les périphériques externes nécessaires en cliquant sur les boutons correspondants dans la barre d'outils des composants, éditer leurs propriétés, spécifier comment ils sont connectés au microcontrôleur et appeler les macros des périphériques.

4. Lancer la simulation pour vous assurer que l'application fonctionne correctement.

5. Transférer l'application dans le microcontrôleur cible par compilation de l'algorigramme, assemblage puis génération du fichier code objet.

II-2) Outils et vues Flowcode

L'environnement de Flowcode consiste en une aire de travail principale dans laquelle s'affiche l'algorigramme, plusieurs barres d'outils qui vous permettent d'ajouter des icônes et des composants à votre application, trois fenêtres spécifiques pour montrer l'état du microcontrôleur ainsi que les composants attachés et enfin deux fenêtres qui montrent les variables et les appels de macros lorsque vous simulez votre application.



Barre d'outils d'icônes
Glissez et déposez une icône de la barre d'outils sur la page de l'algorigramme pour créer votre application.

Barre d'outils composants
Cette barre d'outils propose les composants externes pouvant être connectés au microcontrôleur. Cliquez sur un composant et il sera associé au microcontrôleur dans la vue qui lui est destinée. Les pattes de connexion et les propriétés du composant peuvent être éditées.

Vue du microcontrôleur
Le microcontrôleur actif est montré dans cette fenêtre ainsi que tous les composants externes qui lui sont attachés. Lors de la simulation de l'application, l'état des ports d'entrées/sorties est indiqué à l'aide de couleurs : le niveau haut des sorties est montré en rouge, le niveau bas est affiché en bleu.

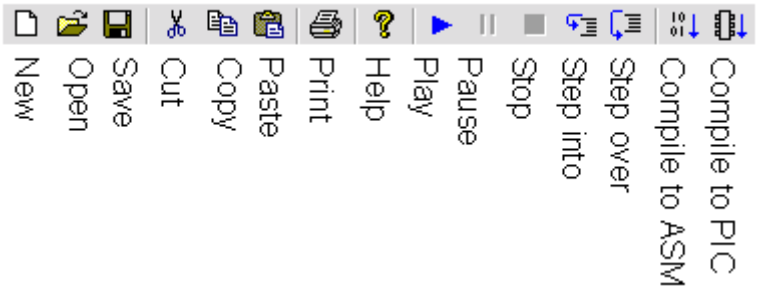
Fenêtre Algorigramme
Les icônes qui constituent l'algorigramme sont affichées dans cette fenêtre. Par contre, les icônes qui constituent une macro sont montrées dans une fenêtre spécifique. La fenêtre principale correspondant à l'algorigramme est toujours visible alors que les fenêtres spécifiques correspondant aux macros peuvent être montrées selon le besoin.

Fenêtre Composant attachés
L'état d'un des composants attachés au microcontrôleur est affiché dans cette vue. Le composant devient 'actif' lors de la simulation de l'algorigramme. Cette vue vous permet aussi d'agir avec des composants externes, par exemple en ouvrant et fermant des interrupteurs.

Fenêtre Variables
Lorsque vous simulez un algorigramme, la valeur de n'importe quelle variable utilisée dans votre application peut être examinée dans cette vue. La valeur des variables est mise à jour lors de toutes les actions, mais cette vue n'est pas rafraîchie lorsque la simulation est faite à la vitesse maximale.

Fenêtre de la pile d'appel
Cette vue affiche le nom de la macro en cours de simulation. Cette vue est très pratique lorsqu'une macro en appelle une autre.

Barre d'outils
Utilisez cette commande pour afficher ou masquer la barre d'outils. La barre d'outils propose des boutons pour quelques-unes des commandes de Flowcode les plus couramment utilisées comme Fichier -> Ouvrir. Vous trouverez aussi sur cette barre d'outils des boutons pour démarrer, arrêter et suspendre la simulation. Une coche apparaît en face de la ligne Barre d'outils dans le menu Affichage pour signaler que la barre d'outils est affichée.



Barre d'état
Utilisez cette commande pour afficher ou masquer la barre d'état tout à fait en bas de la page. Cette barre fournit des explications supplémentaires comme : quelle est l'action exécutée par l'élément de menu pointé, à quoi correspond une icône, etc. Une coche apparaît en face de la ligne Barre d'état dans le menu Affichage pour signaler que la barre d'état est affichée.

II-3) Agrandir les vues

Il est possible d'effectuer un agrandissement de 25 % de la taille normale de la fenêtre de l'algorigramme pour voir une plus grande portion du schéma.
Les fonctions zoom sont aussi accessibles par des touches fonctions.

Vous pouvez voir ci-après le facteur d'échelle disponible ainsi que la touche fonction à utiliser pour l'obtenir:

25%	F2
50%	F3
100%	F4

III) Travailler avec des organigrammes

III-1) Lancer Flowcode

Lors du démarrage de l'application Flowcode, vous avez la possibilité soit de charger un fichier Flowcode existant, soit de créer un nouvel algorigramme.

Flowcode conserve la liste des quatre fichiers les plus récemment utilisés, et ceux-ci peuvent être sélectionnés par un double-clic. Cliquer sur "Autres fichiers…" si le fichier que vous voulez ouvrir ne figure pas dans la liste des fichiers récents.

Problèmes répertoriés : Il arrive que la liste des derniers fichiers utilisés ne fonctionne pas. Lorsque vous choisissez un fichier depuis la liste de lancement de Flowcode ou la liste placée en bas du menu 'Fichier' , il se peut que la compilation ne fonctionne pas correctement. Dans ce cas, ouvrez le fichier à partir de la commande 'Ouvrir' du menu 'Fichier'. Ce problème semble lié aux droits d'accès de l'utilisateur.

III-2) Créer un nouvel organigramme

La commande 'Nouveau' du menu 'Fichier' permet de créer un nouvel algorigramme. Sélectionner le microcontrôleur cible de votre application. Flowcode affichera alors le schéma du microcontrôleur dans la vue du microcontrôleur et créera un algorigramme vide ne comportant que les icônes Début et Fin. Vous pouvez alors commencer à y ajouter vos icônes.

III-3) Ouvrir un algorigramme existant

Il existe plusieurs façons pour ouvrir un algorigramme Flowcode existant.

1. Sélectionner la commande 'Ouvrir' du menu 'Fichier'. Ceci vous permet de rechercher le fichier qui vous intéresse.
2. Sélectionner le fichier depuis la liste des fichiers récemment utilisés accessibles depuis le menu 'Fichier'.
3. Effectuer un double-clic sur un fichier algorigramme depuis l'explorateur de Windows pour démarrer Flowcode et ouvrir ce fichier.

III-4) Sauvegarder un algorigramme

Pour sauvegarder un algorigramme, sélectionner la commande 'Enregistrer' ou 'Enregistrer sous…' du menu 'Fichier'. Les algorigrammes doivent être sauvegardés avant d'être compilés ou transférés dans le microcontrôleur cible.

III-5) Imprimer les algorigrammes

Utiliser la commande 'Aperçu avant impression' du menu 'Fichier' pour afficher l'algorigramme tel qu'il sera imprimé. Quand vous exécutez cette commande, la fenêtre principale est remplacée par la fenêtre de prévisualisation dans laquelle une ou deux pages seront affichées telles qu'elles seront imprimées. La barre d'outils disponible dans l'écran de prévisualisation vous propose des options pour visualiser une ou deux pages à la fois, vous déplacer en arrière ou en avant dans le document; zoomer en avant ou en arrière sur la page et démarrer le travail d'impression.

L’impression peut aussi être obtenue sans passer par la prévisualisation, en sélectionnant directement la commande 'Imprimer' du menu 'Fichier'.

III-6) Fermer Flowcode

Pour fermer un algorigramme Flowcode, sélectionner la commande 'Fermer' du menu 'Fichier'. Pour fermer complètement Flowcode, sélectionner la commande 'Quitter' du menu 'Fichier'. Dans les deux cas, Flowcode vous demandera si vous souhaitez sauvegarder les modifications apportées à votre algorigramme.

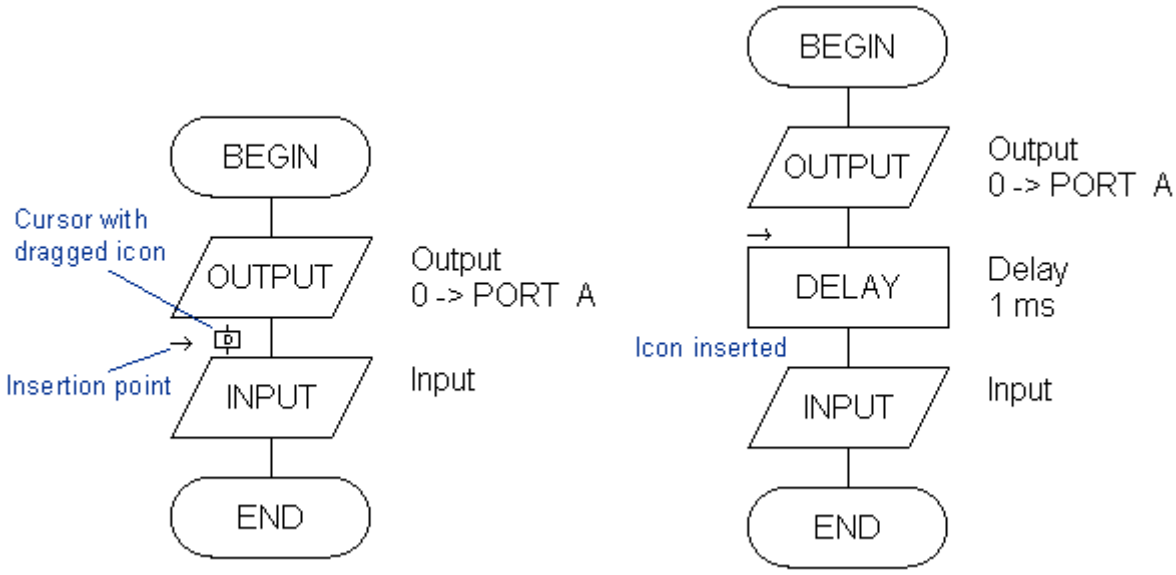
IV) Créer et éditer des algorigrammes

IV-1) Ajouter une icône à un algorigramme

Barre d'icônes

Pour ajouter une icône sur l'algorigramme, cliquez gauche sur celle qui vous intéresse et maintenez le clic. Le curseur prend la forme d'une petite image de l'icône sélectionnée. Faites glisser l'icône dans la fenêtre active de votre algorigramme et relâchez la souris là où vous voulez insérer cette icône. Quand vous déplacez la souris sur l'algorigramme, une petite flèche apparaît pour montrer où sera insérée l'icône quand le bouton de la souris sera relâché. Ce point est identifié comme le point d'insertion.

Dés que vous relâchez le bouton de la souris, l'icône s'inscrit dans l'algorigramme et le nom et les caractéristiques de l'icône apparaissent à sa droite.

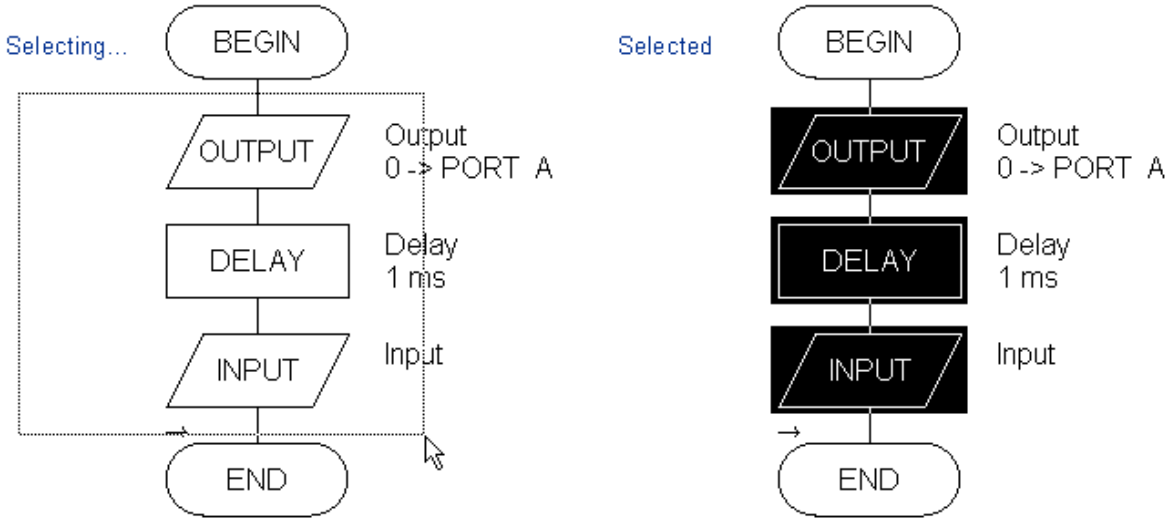


IV-2) Sélectionner les icônes

Avant d'éditer ou de déplacer une icône, il est nécessaire de la sélectionner. Les icônes sélectionnées apparaissent en vidéo inversée sur un fond noir. Il existe deux façons pour sélectionner une ou plusieurs icônes:

1. Cliquer sur une icône pour la sélectionner. Pour l'ajouter ou la soustraire de la sélection il faut maintenir la touche Ctrl du clavier enfoncée pendant le clic.

2. Cliquer et étirer un rectangle autour du ou des icônes à sélectionner.



IV-3) Déplacer, supprimer, copier et coller les icônes

Les icônes peuvent être déplacées, supprimées, copiées dans le presse-papiers, puis collées depuis le presse-papiers.

Pour déplacer une ou plusieurs icônes d'une position vers une autre à l'intérieur du même algorigramme:
Sélectionner les icônes et ensuite cliquer et faire glisser les icônes sélectionnées vers le nouveau point d'insertion. Lorsque vous faites glisser les icônes, le marqueur de point de d'insertion vous indiquera ou les icônes seront insérées lorsque vous relâcherez la souris. Notez qu'il est impossible d'insérer des icônes à un point situé à l'intérieur des icônes sélectionnées.

Pour supprimer des icônes d'un algorigramme:
Sélectionner les icônes à supprimer, puis choisir soit 'Supprimer' soit 'Couper' dans le menu 'Edition'. Si vous utilisez Couper, les icônes seront placées dans le presse-papiers et par conséquent pourront alors être collées dans un autre algorigramme.

Pour copier des icônes depuis un algorigramme:
Sélectionner les icônes puis cliquer sur 'Copier' du menu 'Edition' pour placer une copie des icônes dans le presse-papiers. Ces icônes pourront alors éventuellement être collées dans un autre algorigramme.

Pour coller des icônes depuis le presse-papiers sur un algorigramme:
Choisir le point d'insertion des icônes en cliquant à l'endroit qui convient dans l'algorigramme. Sélectionner ensuite 'Coller' du menu 'Edition'.

Les actions Supprimer, Couper, Copier et Coller sont aussi disponibles dans le menu contextuel qui apparaît lorsque vous cliquez droit sur une icône.

IV-4) Editer les propriétés d'une icônes

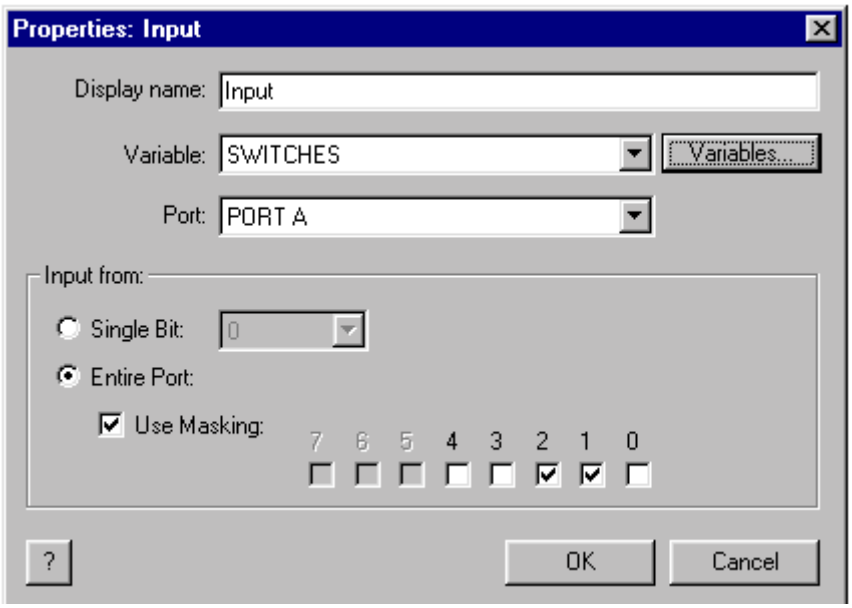
Les propriétés d'une icône définissent son comportement lorsque Flowcode la compile/assemble et la simule. Comme chacune a des propriétés différentes éditables par l'utilisateur, reportez-vous à l'icône qui vous intéresse en particulier.
Pour éditer les propriétés d'une icône, sélectionnez d'abord l'icône sur votre algorigramme puis la commande 'Propriétés' du menu 'Edition'. Vous pouvez plus simplement effectuer un double-clic sur l'icône dans votre algorigramme. Les propriétés d'une icône peuvent aussi être affichées en cliquant droit sur l'icône. Un menu contextuel s'affiche. Vous pouvez alors sélectionner l'option Propriétés de ce menu.

IV-5) Utiliser les masques

Avant d'aborder l'utilisation des icônes, vous devez savoir comment sont utilisés les 'masques' dans Flowcode. Vous souhaiterez peut être revenir à cette page une fois que vous aurez étudié le fonctionnement des icônes Entrée (input) et Sortie (output).

Les "masques" peuvent être sélectionnés pour travailler avec un nombre donné de bits à lire ou à écrire sur un port. Les masques peuvent être utilisés avec les icônes Entrée et Sortie. Les masques sont particulièrement utiles quand un même port dispose à la fois de pattes déclarées en entrées et d'autres en sortie dans la mesure où ils simplifient le contrôle/lecture de un ou plusieurs bits en une seule fois.

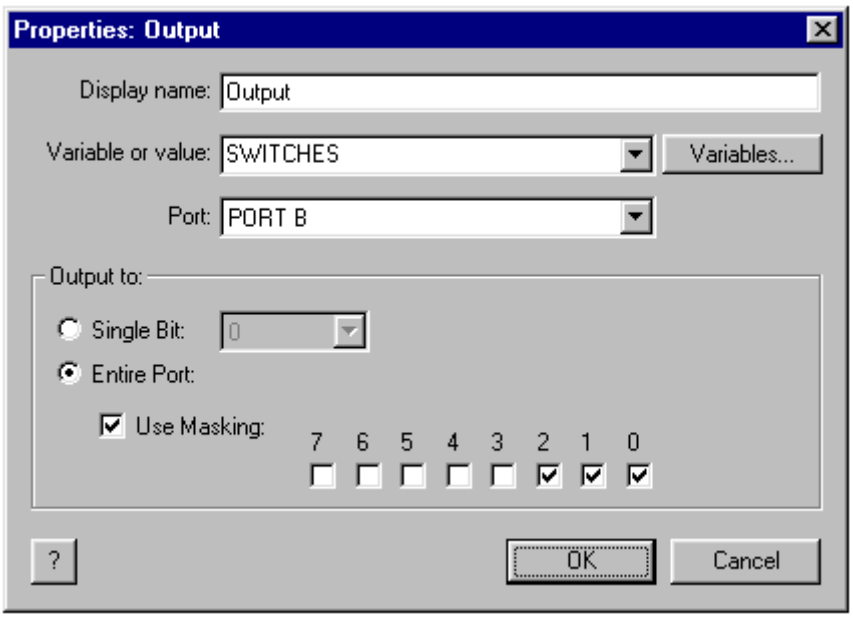
Utilisation de masques en relation avec l'icône Entrée (Input)
Les bits sélectionnés dans un masque font que la valeur correspondante est transmise à la variable. Les bits non sélectionnés retournent zéro.



Exemples:
Imaginons que les 5 Interrupteurs reliés au Port A soient enfoncés. Avec le masque ci-dessus seules les valeurs de A1 et A2 seront passées à la variable SWITCHES donnant la valeur 6, au lieu de 31 correspondant à la somme des 5 bits (1 + 2 + 4 + 8 + 16).

Si l'interrupteur A3 est enfoncé, alors la valeur 0 sera passée à la variable SWITCHES car A3 n'est pas sélectionné.

Utilisation de masques en relation avec l'icône Sortie (Output)
Les bits sélectionnés dans le masque reçoivent la valeur du bit correspondant du nombre ou de la variable envoyée.
Les bits non sélectionnés ne sont pas affectés et conservent leur valeur précédente.

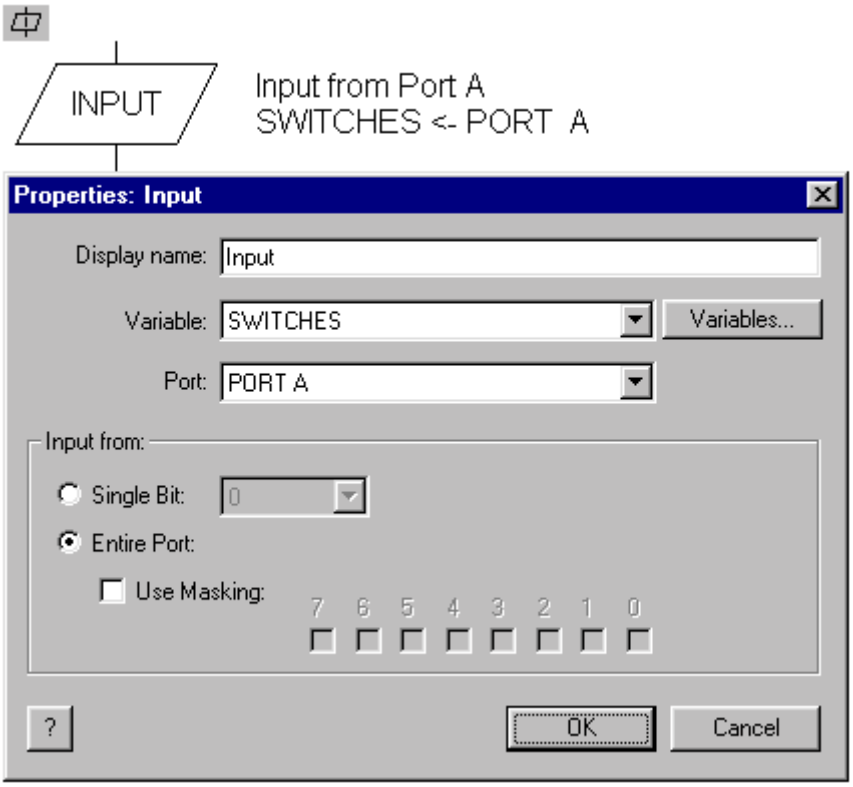


Exemples:
Si SWITCHES vaut 255, et que le masque ci-dessus est appliqué sur cette variable, alors les bits B0 et B2 seront affectés. Les autres bits B3-B7 ne seront pas modifiés et conserveront leur valeur précédente. Les bits B0 - B2 seront mis à 1 puisque les bits correspondants dans la variable SWITCHES le sont aussi.

Si SWITCHES vaut 32 alors les bits B0 à B2 seront mis à zéro puisque les bits correspondant sont à 0.

Le bit B5, qui correspond à la valeur 32, ne sera pas modifié puisqu'il n'est pas sélectionné dans le masque.

IV-6) Propriétés de l'icône Entrée (input)



L'icône Entrée lit le port spécifié (ou certains bits seulement du port) et place le résultat dans la variable fournie.

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

Variable
Sélectionner le nom d'une variable dans laquelle vous souhaitez placer le résultat de la lecture des bits du port.

Bouton Variables
Ce bouton ouvre une boite de dialogue vous permettant de choisir une variable existante ou d'en créer une nouvelle.

Port
Choisir le Port concerné parmi la liste des ports disponibles du microcontrôleur à programmer.

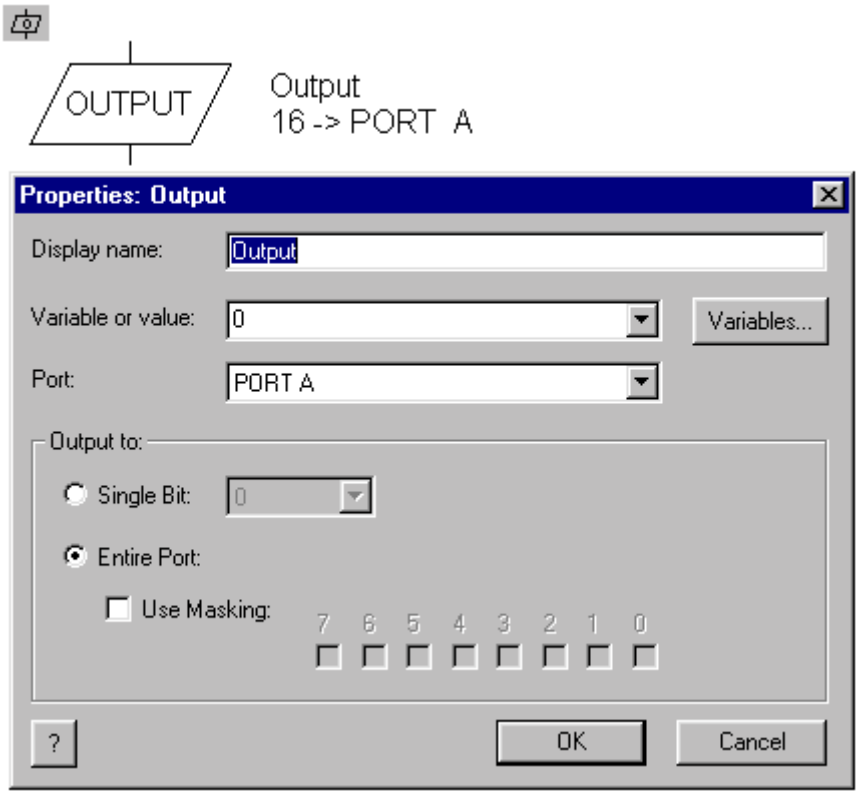
Bit unique
Utilisez cette option pour lire l'état d'un seul bit du port.

Port complet
Utilisez cette option pour lire l'état du port en entier et ranger la valeur lue dans la variable choisie.

Masque
Grâce au masquage, il est possible de lire seulement certains bits dans une variable. Ceci est bien pratique quand certains bits du port sont utilisés en sortie.

Quand un masque est utilisé, seules les valeurs correspondant aux bits sélectionnés sont lues. Reportez-vous à la page Utiliser les masques pour plus d'informations.

IV-7) Propriétés de l'icône Sortie (output)



L'icône Sortie permet d'envoyer la valeur ou le contenu d'une variable au port et/ou aux bits spécifiés.
La sortie est reçue par le port en format binaire.

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

Variable ou valeur
Sélectionner le nom de la variable ou une valeur numérique que vous souhaitez écrire dans ce port.

Bouton Variables…
Ce bouton ouvre la boite de dialogue Variables vous permettant de sélectionner une variable existante ou d'en créer une nouvelle.

Port
Le sélectionner depuis la liste des ports disponibles sur le PICmicro à programmer.

Bit unique
Utiliser cette option pour écrire dans un seul bit du port.
Si une valeur vraie (différente de zéro) est écrite dans ce bit, alors le bit est mis à 1 (mis à ON), sinon le bit est effacé (mis OFF).

Port entier
Utiliser cette option pour écrire la valeur ou la variable dans le port entier.

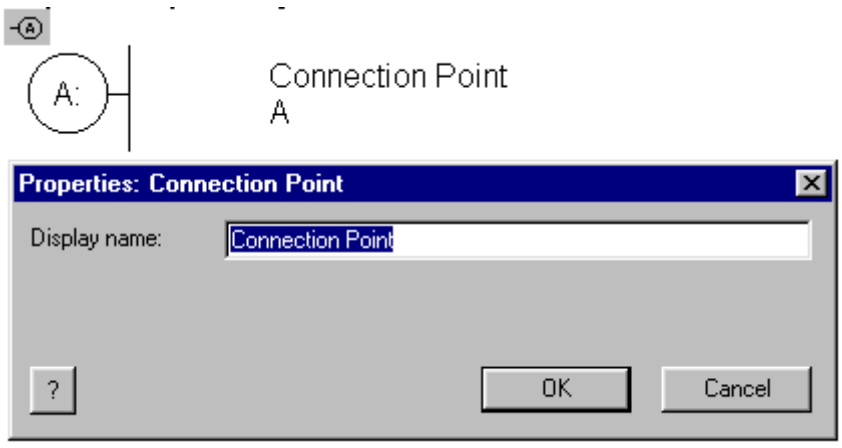
Masque
Il est possible, grâce au masquage, de n'écrire que dans certains bits du port. Ceci est bien pratique quand certains bits du port sont configurés en entrée et que vous souhaitez laisser ces bits inchangés.
Avec le masquage, seuls les bits sélectionnés recevront une valeur. Les autres ne seront pas affectés par cette opération.
Reportez-vous à la page Utiliser les masques pour plus d'informations.

IV-8) Propriétés de l'icône Point de jonction (connection point)

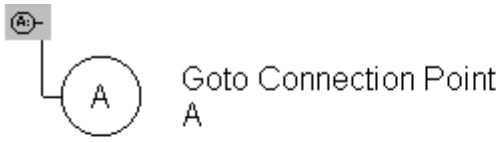
Les icônes point de jonction sont utilisées pour aller d'un endroit de l'organigramme à un autre.
Quand l'organigramme atteint le point de jonction, il saute au point de jonction correspondant et continue ensuite l'exécution à partir de ce point.

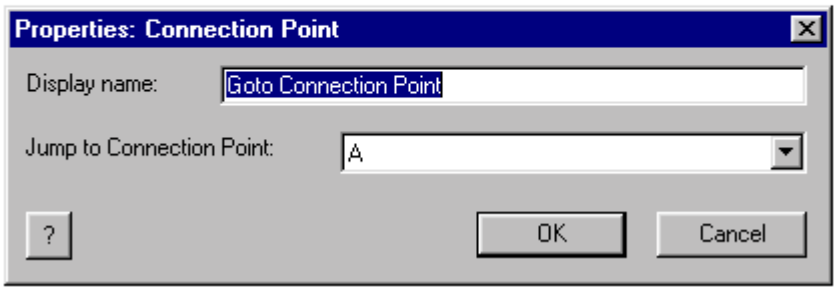
Les icônes de jonction sont utilisées par paires, Le premier est le point de jonction, le point dans l'organigramme indiquant à quel endroit se rendre dans l'organigramme. Le deuxième est le point de saut – le point dans l'organigramme à partir duquel il faut effectuer le saut. Les deux points partagent une lettre de jonction – dans ce cas, la lettre 'A'. Plusieurs points de saut peuvent faire référence à un même point de jonction.

Etape UN: le point de jonction



Etape DEUX: le point de saut

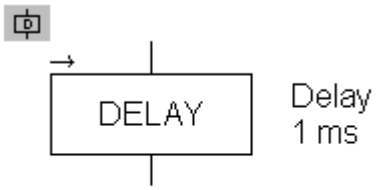


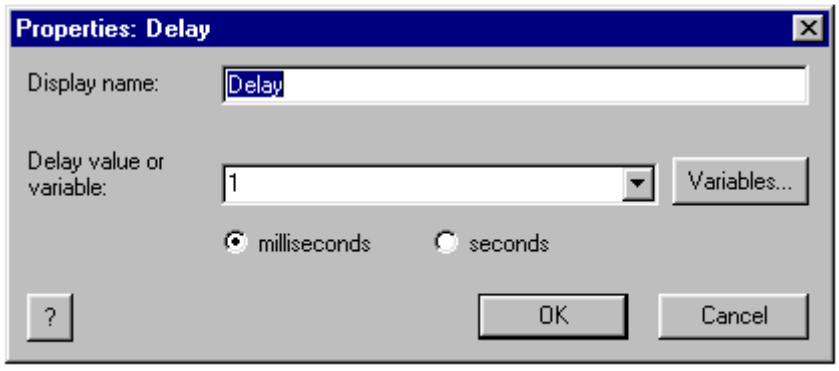


Nom à afficher
Le nom de l'icône à afficher sur l'organigramme.

Aller au point de jonction
Sélectionner le point de jonction auquel vous voulez aller. Cette option n'est pas disponible si l'icône correspond à la définition d'un point de jonction plutôt qu'à un point de saut.

IV-9) Propriétés de l'icône Pause (delay)





Les icônes Pause permettent d'ajuster le timing de votre programme et d'en ralentir l'exécution.
Elles sont particulièrement utiles pour baisser la vitesse d'exécution du programme afin de permettre l'interaction humaine.

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

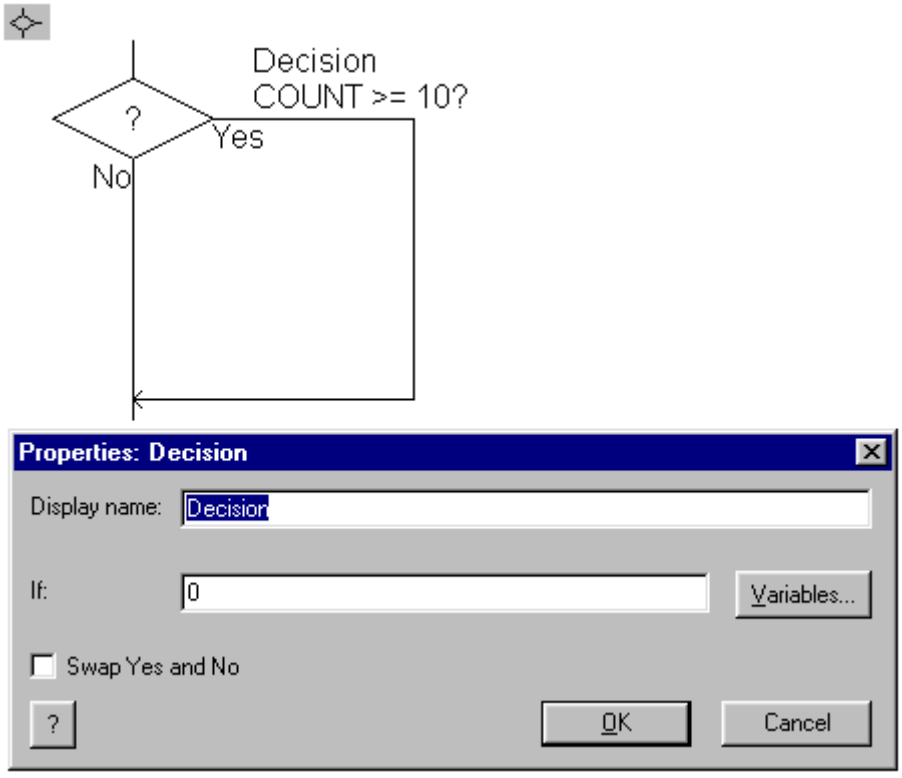
Valeur ou variable Pause
Ceci correspond à la longueur de la pause que vous voulez créer.

Bouton Variables
Ce bouton ouvre la boite de dialogue Variables vous permettant de sélectionner une variable existante ou d'en créer une nouvelle.

Options Millisecondes/Secondes
Les pauses ou temporisations peuvent être exprimées en millisecondes ou en secondes. Lorsque la simulation rencontre une pause exprimée en secondes, une boite de dialogue apparaît montrant le décompte du temps. Le bouton Annuler de cette fenêtre de progression permet de poursuivre l'exécution de l'organigramme sans avoir à attendre que le temps soit complètement écoulé.

Pour permettre une programmation de pause correcte de votre PICmicro, vous devez choisir la vitesse d'horloge adéquate. Pour ce faire utilisez la commande 'Vitesse d'horloge' du menu 'PIC'

IV-10) Propriétés de l'icône Décision (decision)



Les icônes de décision vous permettent de tester une condition et d'effectuer certains traitements en fonction du résultat du test.
Des icônes peuvent être placées dans l'une ou l'autre branche de sortie de l'icône décision.

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

Condition
Le losange Décision teste la condition afin de déterminer dans quelle branche se passera la suite du traitement. Si la condition vaut 0 ou FAUX, c'est la branche 'Non' qui sera déroulée.

Si la condition vaut un nombre différent de 0 ou VRAI alors c'est la branche du OUI qui sera exécutée. Les conditions peuvent contenir des nombres, des variables et des opérateurs comme :

(,)	- Parenthèses.
=, <>	- Egal à, Non égal à.
+, -, *, /, MOD	- Addition, Soustraction, Multiplication, Division & Modulo.
<, <=, >, >=	- Plus petit que, Plus petit ou égal à, Plus grand que, Plus grand ou égal à.
>>, <<	- Décalage à droite, décalage à gauche.
NOT, AND, OR, XOR	- NON, ET, OU, OU Exclusif

A partir du moment ou les variables ont été précédemment définies, toutes les instructions suivantes sont valides.

TEMPO = 10
TEMPO = MAXTEMPO

BITSUIVANT = DERNIERTBIT >> 2 & MASK
AETB = PORT_A AND PORT_B

Bouton Variables
Ce bouton ouvre la boite de dialogue Variables vous permettant de sélectionner une variable existante ou d'en créer une nouvelle.

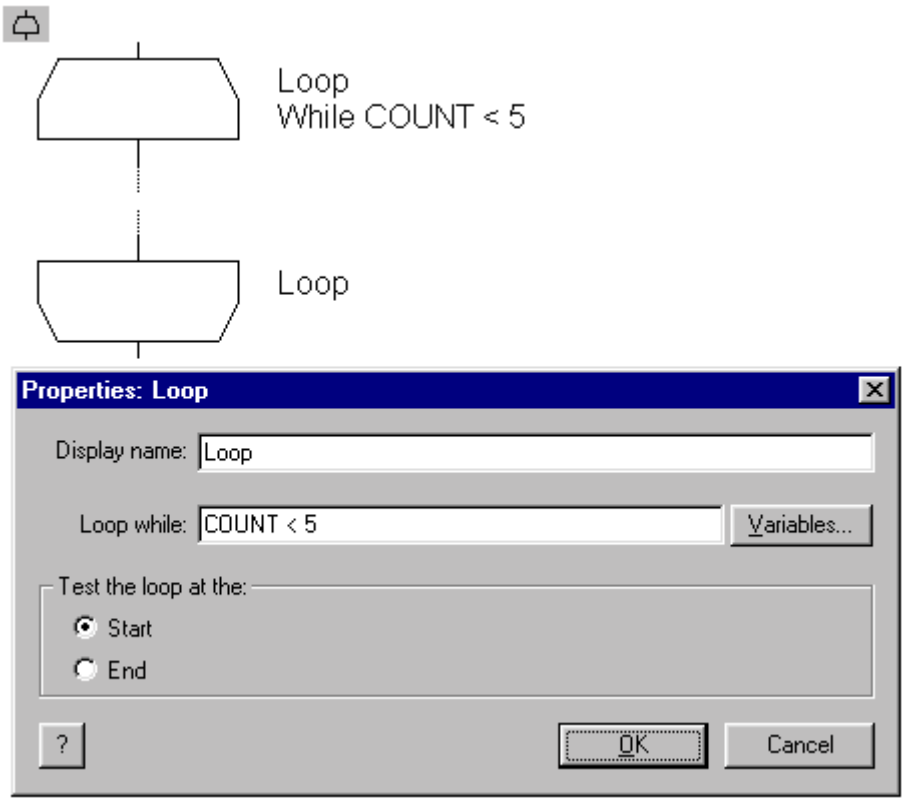
Inverser Oui et Non
Normalement la branche correspondant à "Oui" part sur la droite de l'icône de Décision et la branche correspondant au 'Non' continue tout droit dans l'organigramme. Cocher cette option pour inverser les deux branches.

Valeurs logiques
Flowcode considère zéro comme FAUX et toute autre valeur différente de zéro comme VRAI.

Ceci permet d'utiliser des variables dans un contexte VRAI ou FAUX.

Par exemple, 'If TEMPO_FINI' prendra le chemin 'VRAI' si TEMPO_FINI est différent de zéro.

IV-11) Propriétés de l'icône Boucle (loop)



Les icônes Boucle sont utilisées pour répéter une tâche tant que la condition spécifiée est remplie.
Notez que vous devrez ajouter une icône Calcul qui modifie la variable spécifiée dans la condition pour que cette condition ait une chance d'être satisfaite.

Nom à afficher
Le nom qui apparaîtra sur l'organigramme.

Boucle Tant que (while)
Entrer la condition qui détermine la fin de la boucle.

Définir un test conditionnel toujours Vrai fera que la boucle sera répétée indéfiniment. While 1 en est un exemple.

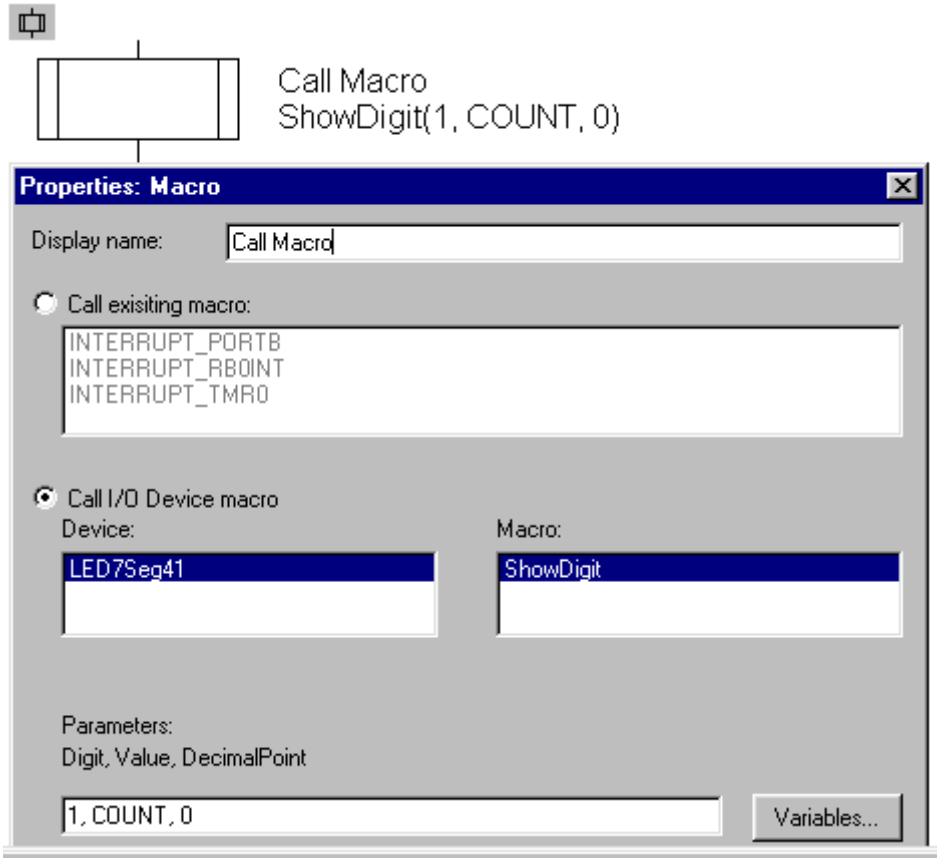
Bouton Variables
Ce bouton ouvre la boite de dialogue des variables vous permettant de sélectionner une variable existante ou d'en créer une nouvelle.

Tester la boucle
Cette option permet de spécifier si la boucle doit être testée au début ou à la fin de la boucle.

Boucle infinie
Il arrive qu'une tâche soit répétée indéfiniment. Une façon pratique d'obtenir ce fonctionnement est d'utiliser une boucle infinie.

Tester une condition toujours Vrai fera que la boucle sera répétée indéfiniment. While 1 en est un exemple.

IV-12) Propriétés de l'icône Macro (macro)



Les Macros sont des portions de code, qui peuvent être utilisées et réutilisées dans différents projets.
Les macros font que des tâches complexes peuvent être gérées par du code pré-écrit. Les macros peuvent être importées et exportées. Dans Flowcode, trois types de macros sont définis :

- Macros interne à Flowcode – par exemple pour les interruptions du PICmicro
- Macros constitués d'organigrammes – en fait un sous-programme dans votre programme
- Macros d'E/S – sous-programmes cachés qui exécutent des fonctions spécifiques pour certains périphériques

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

Appelle une macro existante
Sélectionner cette option pour intégrer une macro de votre crû dans votre organigramme ou pour choisir macro activée suite à une interruption. Toutes les macros existantes figurent dans la liste en dessous de cette option.

Appelle une macro spécifique d'E/S
Sélectionner cette option pour appeler une macro pour un composant externe attaché au microcontrôleur. Sélectionner le périphérique dans la première liste et ensuite sélectionner le nom de la macro que vous voulez appeler dans la seconde liste.

Paramètres
Si la macro exige des paramètres, alors ces paramètres doivent être entrés dans ce champ. Il peut s'agir de valeurs numériques ou de variables existantes. Chaque variable ou valeur doit être séparée par une virgule dans la liste.

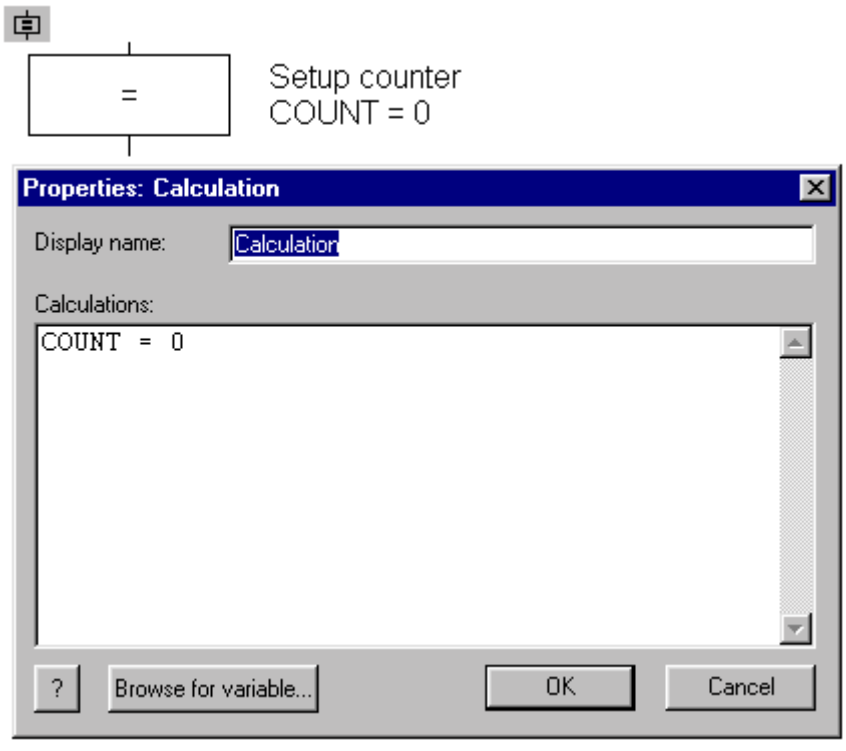
Bouton Variables
Ce bouton ouvre la boite de dialogue Variables vous permettant de choisir une variable existante ou d'en créer une nouvelle.

Valeur de Retour
Si la macro de votre périphérique retourne une valeur, alors vous pouvez affecter cette valeur à une variable existante pour l'utiliser après dans votre organigramme. Si la fonction retourne une valeur que vous ne voulez pas sauvegarder alors laissez ce champ vide.

Bouton Créer une nouvelle macro
Cliquez sur ce bouton pour créer une nouvelle macro dans votre organigramme.

Bouton OK & Editer macro
Après avoir choisi le nom d'une macro existante (non pas une macro pour un périphérique externe), vous pourrez voir et éditer la macro sélectionnée en cliquant sur ce bouton.

IV-13) Propriétés de l'icône Calculs (calculation)



L’icône de Calcul permet la modification des variables. Elle peut être utilisée pour vérifier des entrées ou créer des sorties.

Nom à afficher
Il s’agit du nom de l’icône qui apparaîtra sur l’organigramme.

Calculs
Une ou plusieurs lignes de calculs peuvent être entrées dans cette boite de dialogue.
Tous les calculs doivent comprendre le nom d’une variable existante, le signe égal suivi d’une expression faite de nombres, de variables et des opérateurs suivants :

( )	- Parenthèses.
= <>	- Egal à, Non égal à.
+ - * / MOD	- Addition, Soustraction, Multiplication, Division & Modulo.
< <= > >=	- Plus petit que, plus petit ou égal à, Plus grand que, Plus grand ou égal à.
>> <<	- Décalage à droite, décalage à gauche.
NOT AND OR XOR	- NON (inversion), ET, OU, OU Exclusif

A partir du moment ou les variables ont été précédemment définies, toutes les lignes suivantes sont des lignes de calculs parfaitement valides.

TEMPO = TEMPO + 1
TEMPO = (MA_VARIABLE + 3) * 3
BITSUIVANT = DERNIERBIT >> 2 & MASK
AETB = PORT_A AND PORT_B

Bouton Montrer Variables
Ce bouton ouvre la boite de dialogue des variables vous permettant de choisir une variable existante ou d’en créer une nouvelle.

IV-14) Propriétés de l'icône Interruption (interrupt)

Il est souvent pratique de disposer d’une méthode pour interrompre le microcontrôleur dans sa tâche en cours pour lui faire faire quelque chose de plus important – par exemple répondre au signal d’alarme transmis par le capteur de débordement d’un réservoir ou lorsqu’un certain temps s’est écoulé lorsqu’il s’agit de système fonctionnant sur le temps.

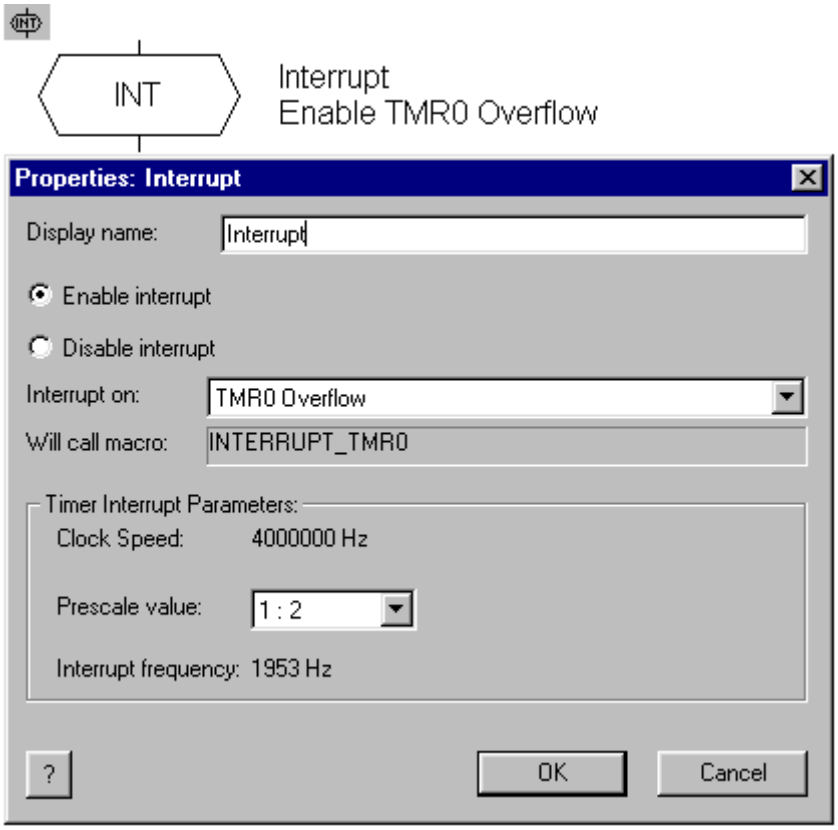
Les microcontrôleurs PICmicro disposent de plusieurs méthodes standards pour provoquer une interruption :

L’interruption dû au débordement du TIMER0 (TMRO overflow), l’interruption déclenchée par le changement d’état d’une des lignes 4, 5, 6 et 7 du port B (RB port change), et un signal reçu (interruption externe) sur la patte B0 du port B (RB0/INT).

Vous pouvez éditer la macro Interruption en sélectionnant la commande 'Edition/Supprimer' du menu 'Macro'. Flowcode affiche alors une liste des macros, incluant celles qui concernent les interruptions. Choisissez le nom de la macro depuis cette liste puis cliquez sur les boutons

Editer ou Supprimer. Si vous choisissez d'éditer la macro, Flowcode ouvrira la fenêtre contenant la macro pour vous permettre de la modifier.

AVERTISSEMENT
L'utilisation de l'interruption externe ou des interruptions dues au changement d'état d'une des pattes RB4 à RB7 du port B implique une certaine connaissance du PICmicro utilisé, puisqu'il y aura des réglages internes assez spécifiques à faire. Reportez vous aux notes techniques de votre PICmicro pour plus de détails.



Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme.

Interruption autorisée/Interruption inhibée
Cocher une des ces options pour autoriser ou inhiber les interruptions.

Source
Utiliser cette liste déroulante pour spécifier la source d'interruption que vous souhaitez autoriser ou inhiber.

Débordement TMR0 – appelée chaque fois qu'un débordement du Timer 0 survient

RB0/INT – appelée lorsqu'un signal survient sur la patte 0 du port B (interruption externe)

Modification Port RB – appelée chaque fois que survient une modification d'une des lignes 4, 5, 6 ou 7 du port B

Macro appelée
Le nom de la macro qui sera appelée lorsque l'interruption surviendra est affiché ici. Le nom de ces macros est n'est pas modifiable. Les macros elles-mêmes ne peuvent être supprimées de l'organigramme bien que vous puissiez placer toute icône de votre choix à l'intérieur de ces macros.

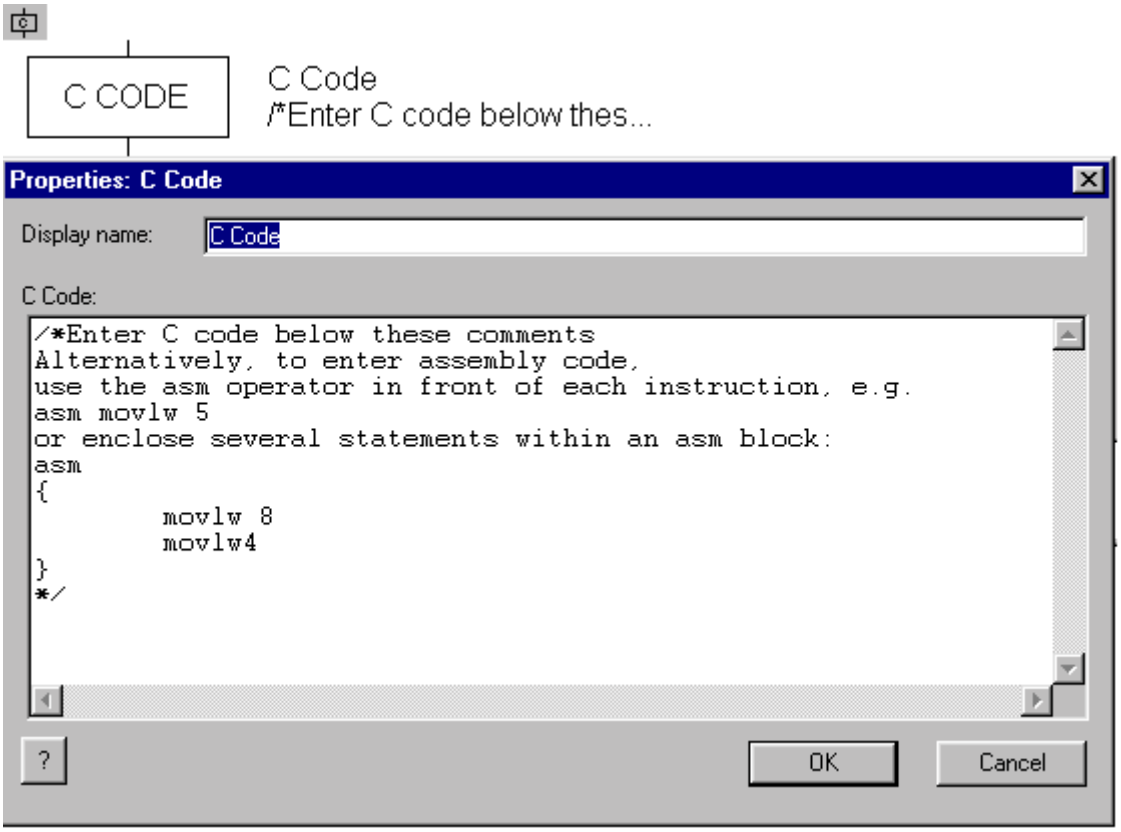
Paramètres de l'interruption Timer :
Si l'interruption suite au débordement du TMR0 est sélectionnée, alors le réglage des paramètres suivants devient possible.

Vitesse d'horloge
Lorsque l'interruption suite au débordement du TMR0 est sélectionnée, la fréquence des interruptions est fonction de la vitesse d'horloge du microcontrôleur et de la valeur du pré-diviseur. La vitesse d'horloge affichée est celle sélectionnée par la commande 'Vitesse d'horloge' du menu 'Exécuter'.

Valeur du pré-diviseur
Régler cette valeur pour déterminer combien de cycles horloges sont nécessaires avant d'incrémenter le compteur du timer0. Utiliser une valeur de pré-diviseur importante pour abaisser la fréquence des interruptions du timer0.

Fréquence d'Interruption
La fréquence des interruptions générées par les débordements du TMR0 est calculée et affichée ici. La fréquence des interruptions sera fonction de la Vitesse d'horloge et de la valeur choisie du pré-diviseur. Cette valeur est exprimée en Hz.

IV-15) Propriétés de l'icône Code (code)



Flowcode est conçu pour permettre à ceux qui débutent dans la programmation des microcontrôleurs PIC de concevoir un projet avec une connaissance limitée des langages de haut niveau.
Cependant, il est possible d'inclure du code écrit dans d'autres langages dans des programmes Flowcode. Des programmes écrits en C et en Assembleur peuvent être inclus avec Flowcode grâce à l'icône Code.
Ceci signifie qu'il est possible de prendre des programmes complexes (vous en trouverez beaucoup sur Internet) écrits en C ou en assembleur et les inclure dans vos projets.

Remarque : Ce code ne pourra pas être simulé par Flowcode, mais sera transmis au microcontrôleur durant la compilation.

Nom à afficher
Le nom de l'icône qui apparaîtra sur l'organigramme

Code C
Entrer le code C que vous souhaitez inclure à votre organigramme. Le code C n'est pas contrôlé par Flowcode mais est transmis directement au compilateur C lorsque l'organigramme est compilé.
Il est important de vérifier que le code C entré est correct, puisque les erreurs éventuelles de syntaxes feront échouer la compilation de tout votre organigramme.

Pour accéder aux variables Flowcode, aux macros et aux points de jonction, il est nécessaire de caractériser l'élément dans votre code C par les préfixes respectifs FCV_, FCM_ et FCC_NomMacro_.

Par exemple, pour utiliser la variable Flowcode appelée TEMPO dans votre code C, vous devrez y faire référence en utilisant FCV_TEMPO. Notez que toutes les variables définies avec Flowcode sont écrites en majuscules.

Pour utiliser la macro Flowcode appelée TEST dans votre programme en C, vous devrez l'appeler FCM_TEST();. Notez que tous les noms de macros Flowcode doivent s'écrire en majuscules.

Pour aller à un point de jonction nommé A, défini dans une macro Flowcode nommée TEST, votre code C doit y faire référence par FCC_TEST_A.. Les points de jonction définis dans l'organigramme principal de Flowcode doivent contenir le préfixe FCC_Main_.

Pour entrer un caractère Tab dans la fenêtre du Code C, utiliser Ctrl+Tab

Code assembleur
Il est possible d'entrer des instructions assembleur dans la fenêtre de Propriétés du code C.

Pour une ligne d'assembleur, utiliser l'opérateur asm devant l'instruction, par exemple :

asm movlw 5

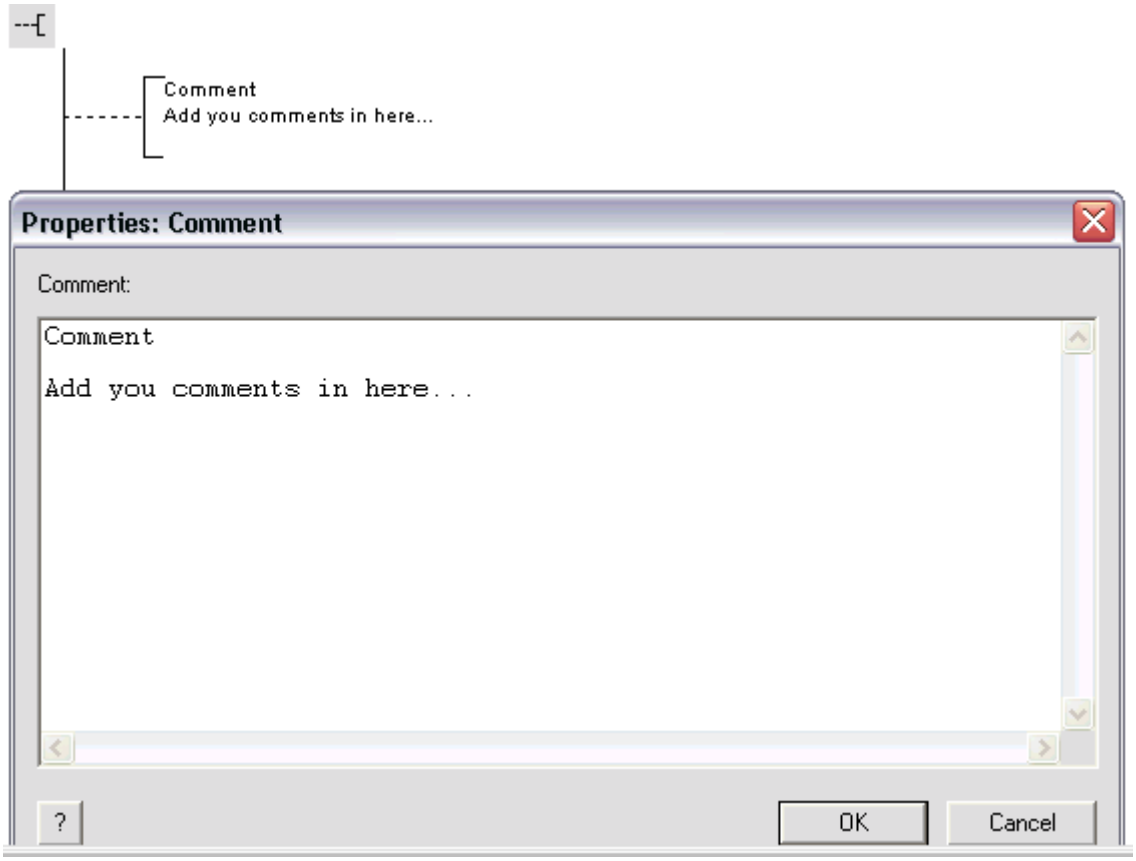
Vous pouvez aussi spécifier plusieurs lignes d'assembleur. Procédez de la façon suivante pour encadrer plusieurs instructions à l'intérieur d'un bloc asm :

asm
{
; Entrer votre code ici
}

Pour accéder aux variables Flowcode ainsi qu'aux macros et aux points de jonction, il est nécessaire de caractériser l'élément utilisé par un préfixe précédé du caractère _ (souligné), à savoir _FCV_, _FCM_ and _FCC_NomMacro_ respectivement.

Les exemples précédents deviennent alors _FCV_TEMPO, _FCM_TEST() et _FCC_TEST_A.

IV-16) Propriétés de l'icône Commentaire (comment)



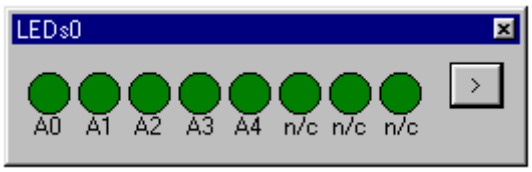
Vous pouvez ajouter des commentaires dans votre code grâce à l'icône Commentaire. Glissez l'icône à l'endroit où vous souhaitez ajouter les commentaires, puis ajouter vos commentaires dans la fenêtre de propriétés.

Notez qu'un commentaire est placé sur le côté de l'algorigramme car il ne fait pas partie du flot d'information à simuler.

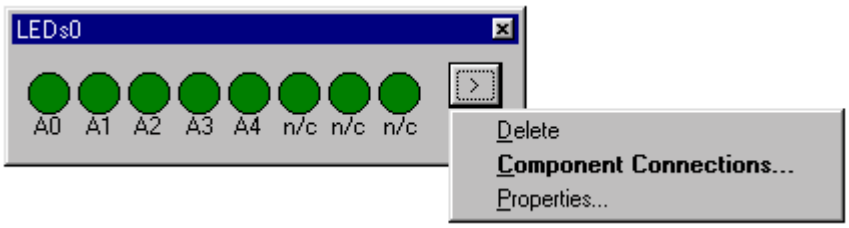
V) Ajouter et éditer des composants

V-1) Ajouter des composants

Pour ajouter un composant externe sur l'algorigramme, cliquez tout simplement sur le bouton approprié depuis la barre d'outils Composants. Flowcode ajoutera un exemplaire de ce composant dans la fenêtre principale.



Pour supprimer le composant externe ou éditer ses connexions ou propriétés cliquer sur le petit bouton ' > ' pour afficher le menu.



Supprimer
Supprime le composant.
Attention : Flowcode ne vous permettra pas de supprimer un composant référencé dans une des macros de votre algorigramme. Supprimer les macros avant de supprimer ce composant.

Connexions...
En sélectionnant cette option, vous pouvez configurer les connexions du composant. Reportez-vous à Editer les connexions du composant pour plus d'informations.

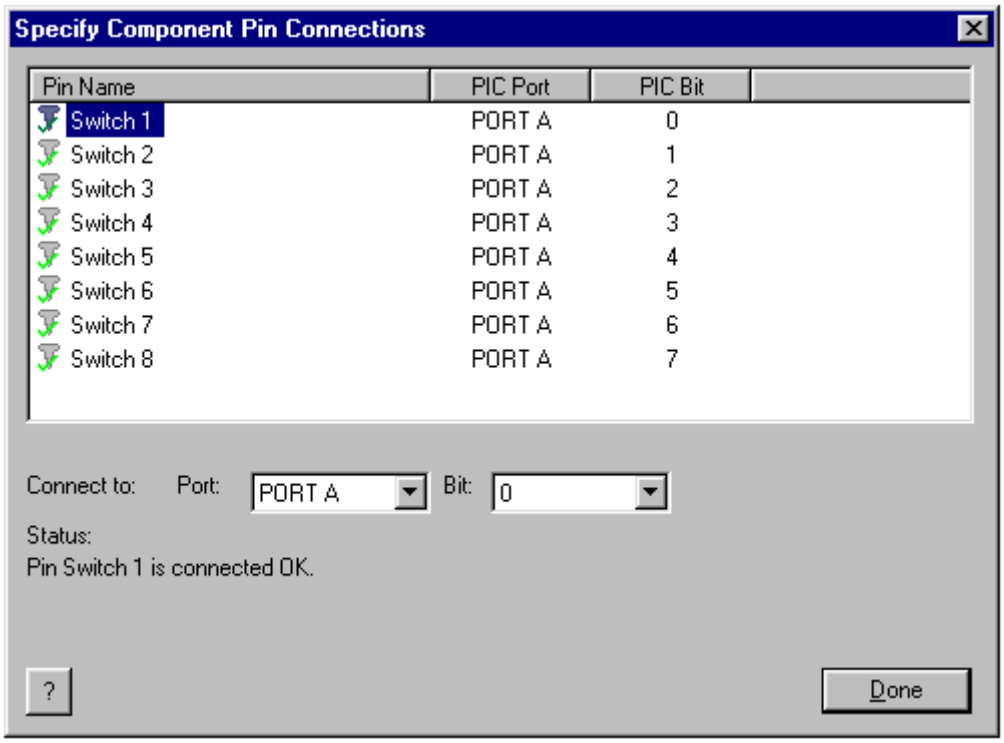
Propriétés...
Certains composants peuvent être configurés à l'aide des propriétés complémentaires qu'ils possèdent. Pour éditer les propriétés d'un composant, cliquer sur Propriétés après avoir cliqué sur le bouton ' > ' du composant. Une boite de dialogue Propriétés du composant s'affiche à l'écran. Si le composant ne dispose pas de Propriétés supplémentaires, alors une page vierge de Propriétés apparaîtra.

Reportez-vous à l'aide spécifique à chaque composant pour de plus amples détails sur chacune de ses Propriétés.

Notez que si vous étudiez un de nos tutoriaux, il se peut que des composants attachés ne soient pas montrés. Sélectionnez alors la commande 'Composants Attachés' du menu 'Affichage'. Une liste des composants attachés est montrée. Cliquez sur le ou les composants que vous voulez faire apparaître dans la fenêtre active.

V-2) Editer les connexions d'un composant

Pour spécifier comment connecter le composant externe au microcontrôleur, afficher d'abord le composant dans la vue du microcontrôleur et cliquez ensuite sur le bouton ' > ' pour faire apparaître un menu contextuel. Cliquez ensuite sur Connexions. La boite de dialogue pour la connexion des pattes du Composant s'ouvre en montrant l'état actuel des connexions et autorise leurs modifications.



Nom des pattes
Le nom des pattes du composant externe est montré dans une colonne. Chacune étant précédée d'une image indiquant l'état de la connexion. Une coche verte signale que la connexion est OK. Une croix rouge indique que la patte n'est pas connectée au microcontrôleur mais doit l'être pour que le composant fonctionne correctement. Un point d'interrogation jaune indique que la patte n'est pas connectée mais que ce n'est pas essentiel pour un fonctionnement correct du composant.

Port
La patte du port du microcontrôleur sur laquelle est connectée une patte externe est affichée dans cette colonne.

Bit
Le bit du port du microcontrôleur auquel est connectée une patte externe est affiché dans cette colonne.

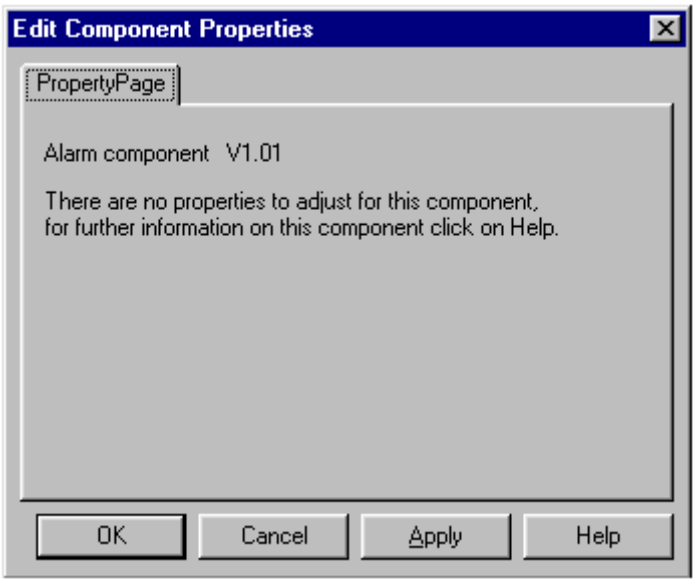
Etat
Les informations d'état d'une patte sont affichées ici. Vous pourrez y voir éventuellement des détails sur des conflits possibles pour des composants utilisant les mêmes pattes du microcontrôleur. De plus, certains composants nécessitent de connecter certaines pattes sur le même port avec d'autres pattes du composant externe. Dans ce cas, la ligne d'état indiquera cette information et l'utilisateur ne pourra apporter aucune modification dans la zone Connecter.

Connecter
Utiliser cette rubrique pour déterminer la patte du microcontrôleur à utiliser. Si cette rubrique est indisponible, c'est que la connexion de la patte est déterminée par celle d'une autre patte. Un message apparaîtra dans la ligne d'état juste en dessous si c'est le cas.

Pour modifier la connexion, cliquer d'abord sur la patte à modifier depuis la liste affichée dans la colonne Nom des pattes.

V-3) Fichier d'aide des composants

Des fichiers d'aide spécifique à chaque composant peuvent être consultés via l'écran Propriétés. Le bouton d'aide ' ? ' figurant dans cette boite de dialogue vous permet d'accéder directement à l'aide de ce composant en particulier plutôt qu'au fichier d'aide principal de Flowcode.



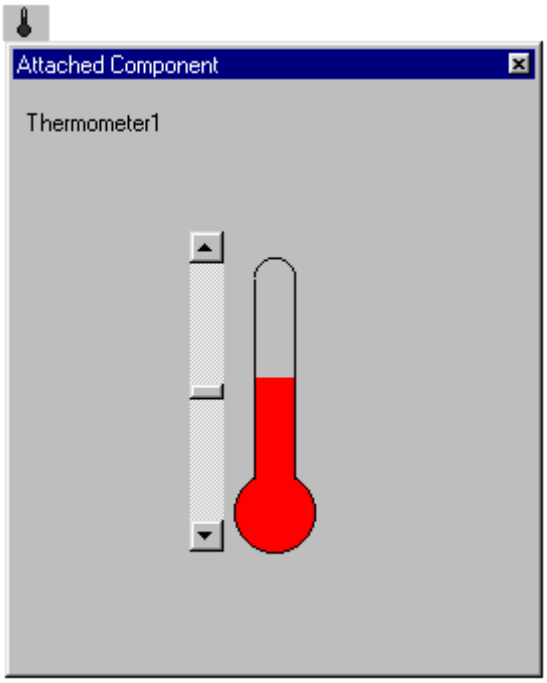
Ceci permet l'ajout de composants complémentaires - à une date ultérieure ou provenant de tiers - ayant un fichier d'aide accessible depuis Flowcode.

Le fichier d'aide décrit en détails les connexions et le mode de fonctionnement du composant.

Des fichiers d'aide pour les composants standards ont aussi été incorporés à ce fichier d'aide principal pour guider l'utilisateur.

V-4) Composants standards

Thermomètre analogique



Le thermomètre est un composant analogique et nécessite un microcontrôleur disposant de un ou plusieurs convertisseurs d'entrée analogique - numérique.
Ce composant enregistre les variations de tension par rapport à une tension de référence. La tension résultante est enregistrée en mémoire.
Les PICmicros sauvegardent la valeur soit sur 8 bits dans un registre, soit sur 10 bits dans deux registres. Dans ce cas, le registre haut contient les 8 bits les plus forts, et le registre bas contient les 2 bits faibles.
Les choix entre 8 et 10 bits dépend du composant utilisé.
Les macros utilisées pour enregistrer les valeurs sur 8 bits ou 10 bits sont les mêmes; mais dans le cas de 8 bits, la macro ReadLowBits retourne toujours 0.

Macros
Le composant thermomètre dispose des macros suivantes:

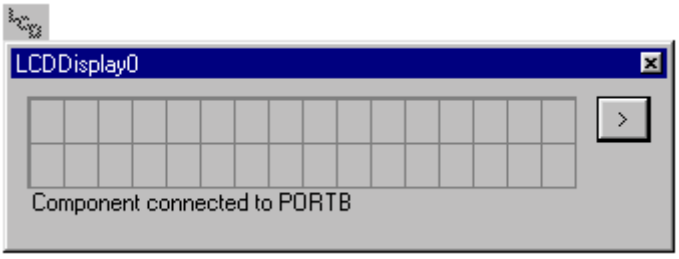
SampleADC
Le PICmicro échantillonne l'entrée analogique et envoie une valeur numérique sur 10 bits aux registres

ReadLowBits
Lit les 2 bits faibles de la valeur sur 10 bits.

ReadHighBits
Lit les 8 bits forts de la valeur sur 10 bits.

Quand vous travaillez avec des entrées analogiques, vous devez faire attention. Les valeurs sont rangées dans deux emplacements mémoire et chaque macro ReadLowBits et ReadHighBits ne retourne qu'une partie de la donnée stockée.

Afficheur LCD



Le composant Afficheur LCD de Flowcode fournit un ensemble de macros simples pour permettre l'affichage de 2 lignes de 16 caractères sur un écran LCD. Ces macros masquent une grande partie de la complexité de programmation d'un afficheur LCD. L'afficheur LCD est un afficheur intelligent de 2 lignes de 16 caractères alphanumériques. Il s'agit d'un composant assez standard, mais il se peut que le jeu de caractère diffère légèrement d'un afficheur à l'autre. Flowcode présume que vous utilisez un afficheur basé sur le contrôleur Hitachi HD44780 (qui est un standard courant dans l'industrie) – et vous ne devriez pas rencontrer trop de problèmes à l'incorporer dans vos différents projets.

Macros
Le composant LCD dispose des macros suivantes:

Start()
La macro init doit être appelée pour initialiser l'afficheur LCD avant d'utiliser toute autre macro se rapportant à cet afficheur.

Clear()
Cette macro efface l'afficheur.

Cursor(x, y)
Cette macro positionne le curseur à la position spécifiée par les coordonnées x et y. Les macros PrintASCII et PrintNumber utilisent la position courante du curseur pour écrire.

PrintNumber(Nombre)
Affiche le nombre fournit. Transmettre le nombre 34 affichera "34" sur le LCD. Lorsque le caractère a été affiché, la position courante du curseur est actualisée automatiquement.
Le nombre doit être compris entre 0 et 255.

PrintASCII(caractères)
Affiche les caractères ASCII à partir de la position courante. Lorsque les caractères ont été affichés, la position courante du curseur est actualisée automatiquement

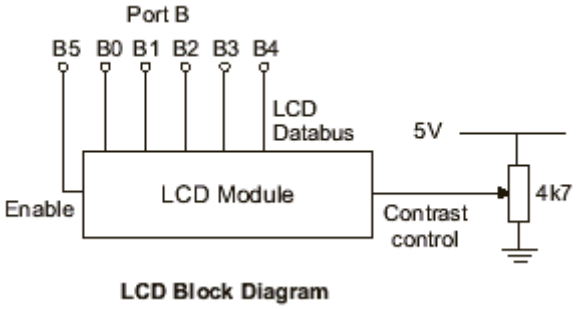
Les caractères peuvent être soit un code caractère ASCII, une chaîne de un ou plusieurs caractères entourés de guillemets telle que "Bonjour", ou un caractère ASCII entouré d'apostrophes tel que 'A'.

NOTE : Seuls les caractères majuscules sont reconnus lors de l'utilisation de caractères entourés par des apostrophes, tel que 'B'.

L'afficheur LCD utilise les caractères ASCII lors de la simulation. Cependant, dans la réalité, les caractères affichés seront fonction du jeu de caractère spécifique à l'afficheur utilisé.
Vous trouverez une liste des caractères ASCII affichable avec la carte de développement de Matrix Multimedia à la fin de ce paragraphe.

Exemples
PrintASCII("Hello") affiche les lettres Hello
PrintASCII("H") affiche la lettre H
PrintASCII('H') affiche la lettre H
PrintASCII('=') affiche le caractère =
PrintASCII(72) affiche la lettre H car 72 (44 en hexadécimal) est le code caractère ASCII de H

Schéma du circuit



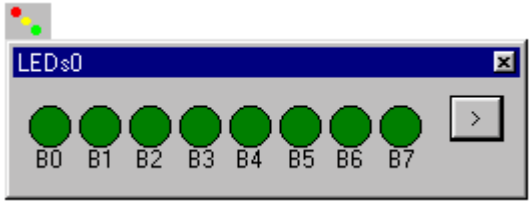
LCD Block Diagram

L'afficheur LCD doit être connecté au microcontrôleur comme montré sur le diagramme.
Typiquement ce périphérique doit être connecté au microcontrôleur via les ports suivants.

Data1:	B0
Data2:	B1
Data3:	B2
Data4:	B3
RS:	B4
Enable:	B5

Table de caractères ASCII de l'afficheur de la carte de développement V2 Matrix Multimedia

LEDs



Le modèle LED de Flowcode permet de créer une rangée de 1 à 8 LEDs colorées. Des macros sont disponibles pour allumer ou éteindre les LEDs individuellement.

Propriétés
Le composant LED dispose de deux propriétés qui peuvent être ajustées.

Nombre de LEDs
Utiliser la liste déroulante pour sélectionner un nombre compris entre 1 et 8 correspondant au nombre de LEDs de la rangée.

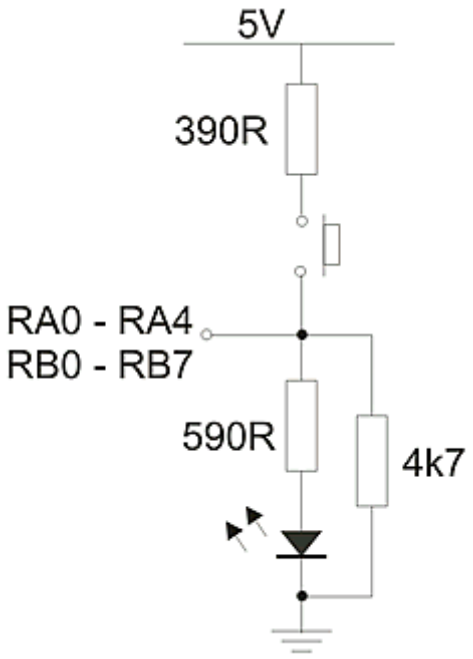
Couleur des LEDS
Utiliser la liste déroulante pour sélectionner la couleur de toutes les LEDs de la rangée. Les couleurs disponibles sont rouge, vert, jaune et bleu.

Macros
Le composant LED dispose de deux macros:

LEDOn (num_LED)
Cette macro allume la LED spécifiée par le paramètre num_LED.

LEDOff (num_LED)
Cette macro éteint la LED spécifiée par le paramètre num_LED.

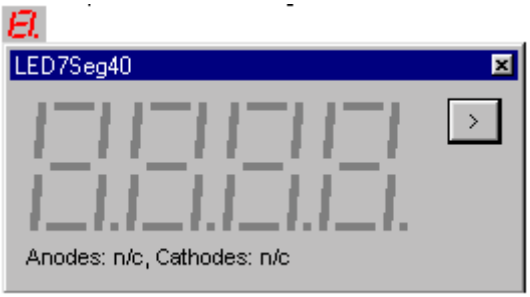
Schéma du circuit



Circuit combinant LED et interrupteur
Ce circuit principal combinant à la fois des LEDs et des interrupteurs vous permet d'utiliser soit un interrupteur, soit une LED sur une patte d'E/S du micro PIC. Dans ce composant, les LEDs sont câblées de la façon suivante : la cathode est connectée à la masse et l'anode est connectée au port qui convient via une résistance de 560 ohms. Une résistance de 4k7 est utilisée pour ramener au niveau logique 0V la patte quand elle est utilisée comme entrée. La résistance de 390 ohms est utilisée pour limiter l'intensité du courant quand l'interrupteur est fermé.

Quadruple afficheur 7 segments

L'afficheur 7 segments est disponible soit sous la forme d'afficheurs 7 segments individuels, soit sous la forme d'un quadruple afficheur 7 segments.

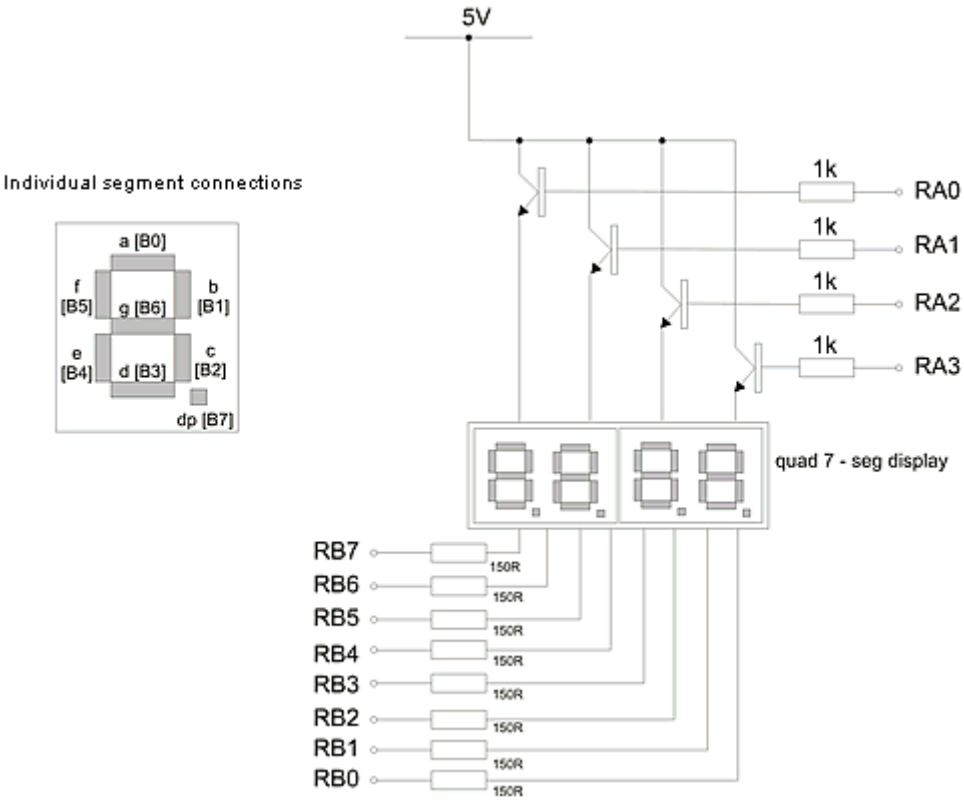


Le composant afficheur à LED 7 segments dispose d'une seule macro pour contrôler l'affichage. Afin de réduire le nombre de connexions avec le microcontrôleur, l'afficheur doit être multiplexé. Dans ce cas chaque afficheur doit être montré successivement afin de donner l'illusion que les 4 digits sont affichés en même temps.

Macros
Le composant quadruple afficheur à 7 segments dispose de la macro suivante:

ShowDigit(Digit, Valeur, Point décimal)
Affiche un nombre sur un l'afficheur spécifié dans le paramètre Digit. Cette valeur doit être comprise entre 1 et 4. Le dernier chiffre du nombre contenu dans le paramètre Valeur est affiché. Par exemple, si Valeur contient 48 alors c'est le chiffre 8 qui sera affiché. Si le paramètre Point décimal contient autre chose que zéro, alors un point décimal sera affiché à la droite du chiffre.

Schéma du circuit

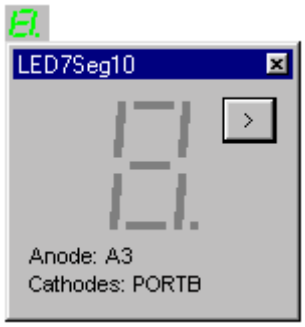


Le quadruple afficheur 7 segments doit être connecté au microcontrôleur comme montré dans le diagramme ci-dessus. Typiquement ce périphérique doit être connecté au microcontrôleur via les ports suivants :

Anode1: A3
Anode2: A2
Anode3: A1
Anode4: A0
SegmentA: B0
SegmentB: B1
SegmentC: B2
SegmentD: B3
SegmentE: B4
SegmentF: B5
SegmentG: B6
SegmentDP: B7

Mono afficheur 7 segments

L'afficheur 7 segments est disponible soit sous la forme d'afficheurs 7 segments individuels, soit sous la forme d'un quadruple afficheur 7 segments.



Macros
Le composant afficheur à 7 segments dispose de la macro suivante:

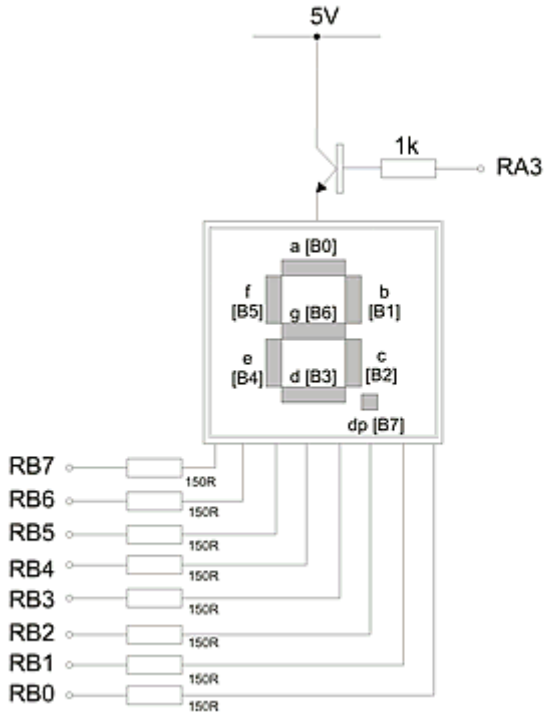
ShowDigit(Valeur, Point décimal)
Affiche un nombre sur l'afficheur. Le dernier chiffre du nombre contenu dans le paramètre Valeur est affiché. Par exemple, si Valeur contient 48 alors c'est le chiffre 8 qui sera affiché. Si le paramètre Point décimal contient autre chose que zéro, alors un point décimal sera affiché à la droite du chiffre.

Schéma du circuit
L'afficheur 7 segments doit être connecté au microcontrôleur comme montré dans le diagramme ci-dessous. Typiquement ce périphérique doit être connecté au microcontrôleur via les ports suivants :

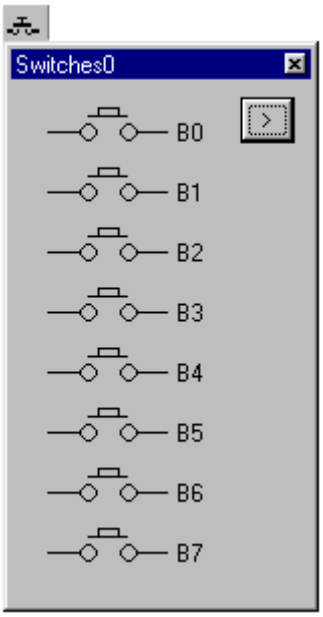
Anode: A3

SegmentA: B0
SegmentB: B1

SegmentC: B2
SegmentD: B3
SegmentE: B4
SegmentF: B5
SegmentG: B6
SegmentDP: B7



Interrupteurs



Le composant Interrupteurs de Flowcode permet de connecter une rangée de 1 à 8 interrupteurs au microcontrôleur.
Il existe une macro pour lire l'état de chaque interrupteur individuel.

Il est possible de d'associer à chaque interrupteur une des touches 0 à 9 du clavier numérique pour l'activer lors de la simulation du pavé numérique.

Propriétés
Le composant Interrupteur dispose de deux propriétés qui peuvent être ajustées.

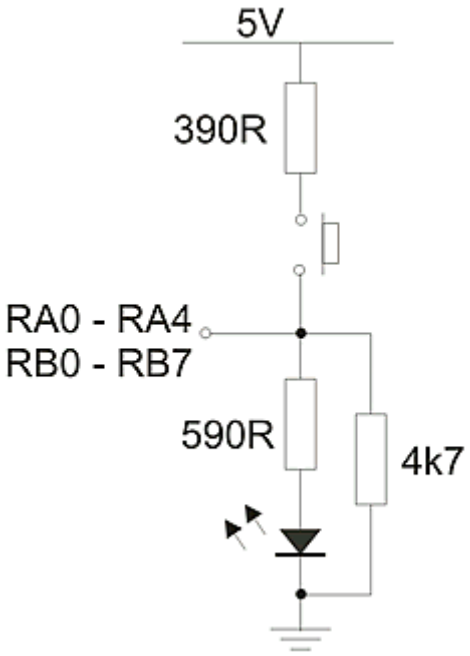
Nombre d'interrupteurs
Utiliser la liste déroulante pour spécifier un nombre d'interrupteurs compris entre 1 et 8 représentant le nombre d'interrupteurs dans la rangée

Type d'interrupteur
Utiliser le menu déroulante pour positionner les switches en position ouverte ou fermée.

Macros
Le composant interrupteur dispose d'une seule macro:

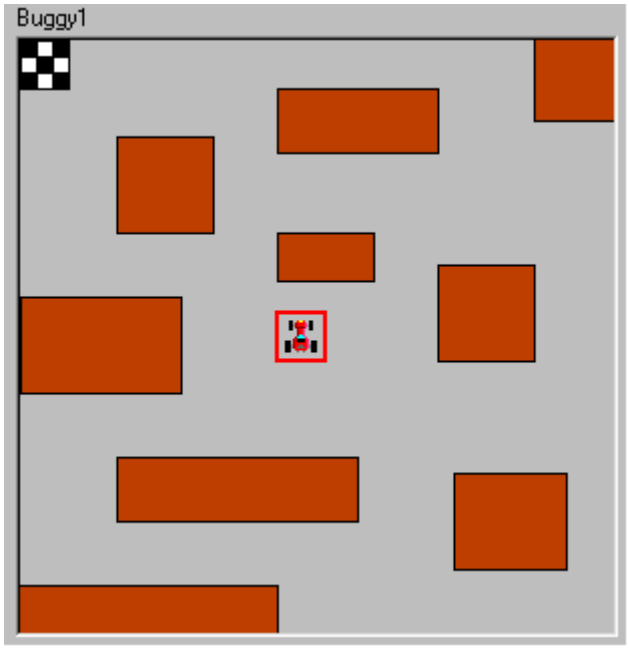
ReadState (num_inter)
Cette macro lit l'état de l'interrupteur spécifié par le paramètre num_inter. Elle retourne 0 si l'interrupteur est ouvert et 1 s'il est fermé

Schéma du circuit



Circuit combinant LED et interrupteur
Ce circuit principal combinant à la fois des LEDs et des interrupteurs vous permet d'utiliser soit un interrupteur, soit une LED sur une patte d'E/S du micro PIC. Dans ce composant, les LEDs sont câblées de la façon suivante : la cathode est connectée à la masse et l'anode est connectée au port qui convient via une résistance de 560 ohms. Une résistance de 4k7 est utilisée pour ramener au niveau logique 0V la patte quand elle est utilisée comme entrée. La résistance de 390 ohms est utilisée pour limiter l'intensité du courant quand l'interrupteur est fermé.

Buggy



The aim is to develop routines for steering the buggy with the ultimate goal of getting through the maze to the chequered flag finishing point.
However the maze can be taken off to give a nice open area for driving manoeuvres.
At the bottom left of the buggy screen is a help button which will open this help file.

The buggy can be steered left and right, and driven forwards or in reverse.
There are front and rear bumper sensors.

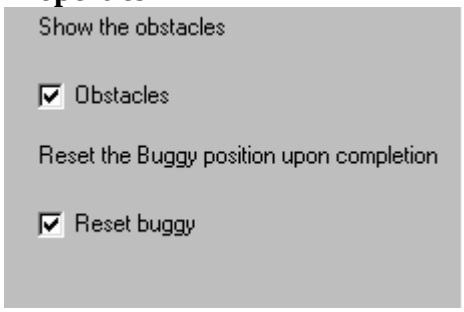
Connections

The three control outputs are on Port A.
The three sensors are on Port B

Pin Name	**PIC Port and Pin**	**Description**
Left wheel	A0	Steers to the left
Right wheel	A1	Steers to the left
Both wheels	A0 and A1	Drives forwards
Reverse	A2	Reverses the buggy
Front Bumper sensor	B0	Obstacle ahead - prevents forwards movement
Rear Bumper sensor	B1	Obstacle behind the buggy - prevents reversing
Objective reached	B2	The chequered objective has been reached.

Note:
To move forwards you need to engage both the left and the right wheel.
If a bumper is activated then the buggy will not be able to move in that direction.

Properties



The properties page has the following options:

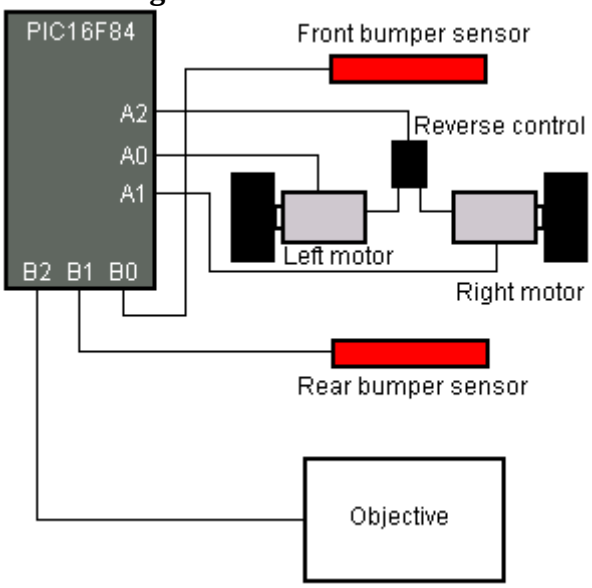
Show the obstacles

· Turns the obstacle maze on or off

Reset the Buggy position upon completion

· Returns the buggy to the centre start box upon completion, and resets to the original heading.

Circuit diagram



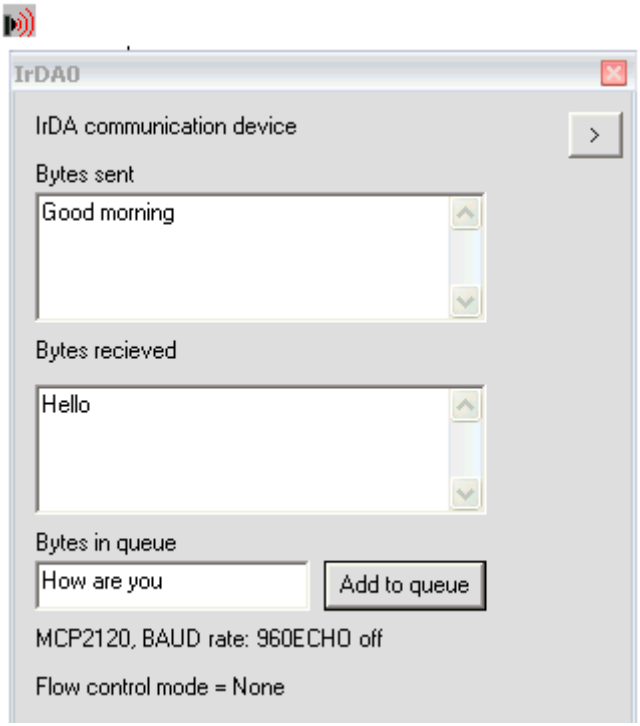
The diagram below gives details of the Input/Output connections for the buggy type envisioned in the component.

The exact method of connection for the inputs will be determined by the buggy used (cable, radio etc.). If the buggy varies from the component you will need to adapt the circuit, and maybe even the code as appropriate.

Please note as well that the buggy envisioned in the component has a reverse, and cut offs to prevent movement when a bumper sensor is activated. The buggy that you use may not have these features.

The objective. The buggy component sends a signal when the objective is reached. To implement this you will need to devise a switch of some sort that can register when the buggy has reached it and send a signal. Examples are: breaking a light beam, or hitting a bumper switch

IrDA



The IrDA component is a communications device used to communicate between the PICmicro and another IrDA equipped device.
This could be another PICmicro with an IrDA connection, a Palm, or any other device that can communicate via IrDA.
Note that IrDA hardware needs to be connected to the PICmicro development board for the IrDA code to function when downloaded to the PICmicro.
This component was designed to function in conjunction with the Matrix IrDA E-block EB-012.

For more information on setting up and using the IrDA E-block please refer to the documentation that accompanied the IrDA E-block.
Or visit the Matrix Multimedia website at www.matrixmultimedia.com for the latest documentation and updates.

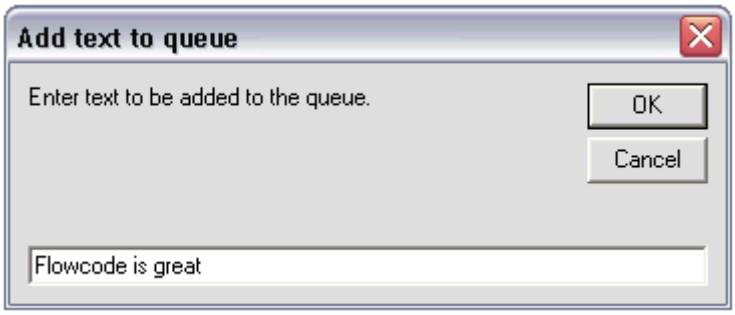
Macros
The Add Defines component has the following macros:

SendIrDAByte(byte)
Sends the byte value via the IrDA connection

char ReceiveIrDAByte(timeout)
Receives the next byte of data from the IrDA connection.
Timeout specifies how long to wait for a byte of data.
The macro returns the data value or 255 if there was no data received.

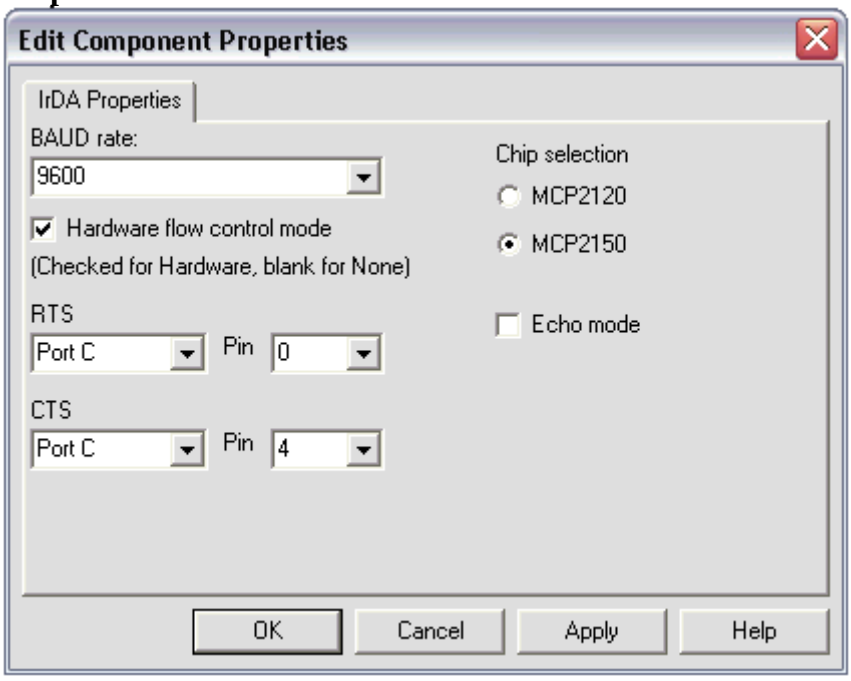
Simulating the IrDA component
The IrDA component simulates IrDA communications as follows:
The component displays any text sent from your program in the "Bytes received" box.
The component sends bytes from the text that is in the "Bytes in queue" box to your program or 255 if no data is present.

Bytes sent are also put into the "Bytes sent" box. this allows you to see what has been sent, as well as what is ready to send.



Enter the text into the box and press OK to add the text to the queue.

Properties



The IrDA component has the following properties:

Baud rate
Sets the communications rate for the IrDA component.
Note: for accurate communications both the sending and receiving units should be operating at the same baud rate.

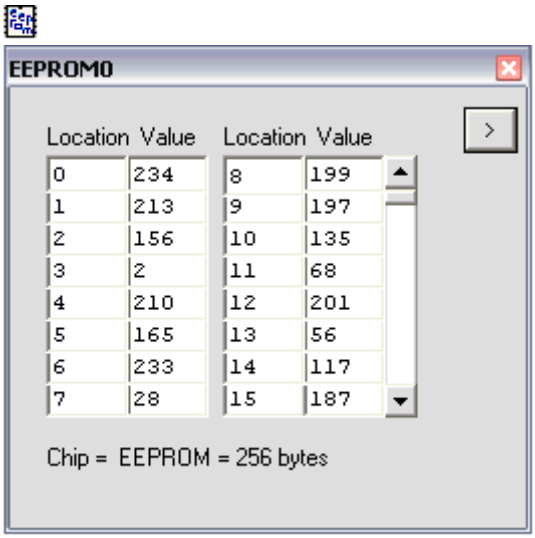
MCP2120/MCP2150 chip select
Selects which IrDA chip to use with the program.
Note that the hardware flow control options are only available with the MCP2150 chip.

Hardware flow control
Enabled Hardware flow control.
For information on flow control settings please consult the documentation on the relevant PICmicro device.

CTS and RTS settings
Allows users to set with Port and which Pin to use for CTS and RTS connections.
Note the CTS and RTS settings are disabled and are not shown on the component when Hardware control is not selected.
For information on the CTS and RTS settings please consult the documentation on the relevant PICmicro device.

Echo mode
Enables ECHO mode.

EEPROM



The EEPROM component allows users to write to and read from the EEPROM memory.

The paired cells show the EEPROM memory location, and the data stored within.
You can scroll up or down to reach particular locations using the scroll bar.
The information text at the bottom contains the target device and the amount of EEPROM memory on that device.

Manually changing memory values
EEPROM data values can be altered manually by double-clicking the data cell and entering the new value.
Important: this is intended for simulation testing only, the EEPROM values are not sent to the PICmicro with the program.

Note: for simulating EEPROM the cells are initialised with random numbers.
This helps reflect that fact that the EEPROM can contain stored values from previous use.

Note: not all PICmicro's have EEPROM memory, and those that have can have various amounts of EEPROM memory onboard.
The component will configure itself accordingly if it detects how much EEPROM the target device has, or if it detects a device without EEPROM.

However, this auto-configuration is limited to those devices supported when the component was created.
You can also manually set how much EEPROM a device has. This allows the EEPROM component to be used with other newer PICmicro's.

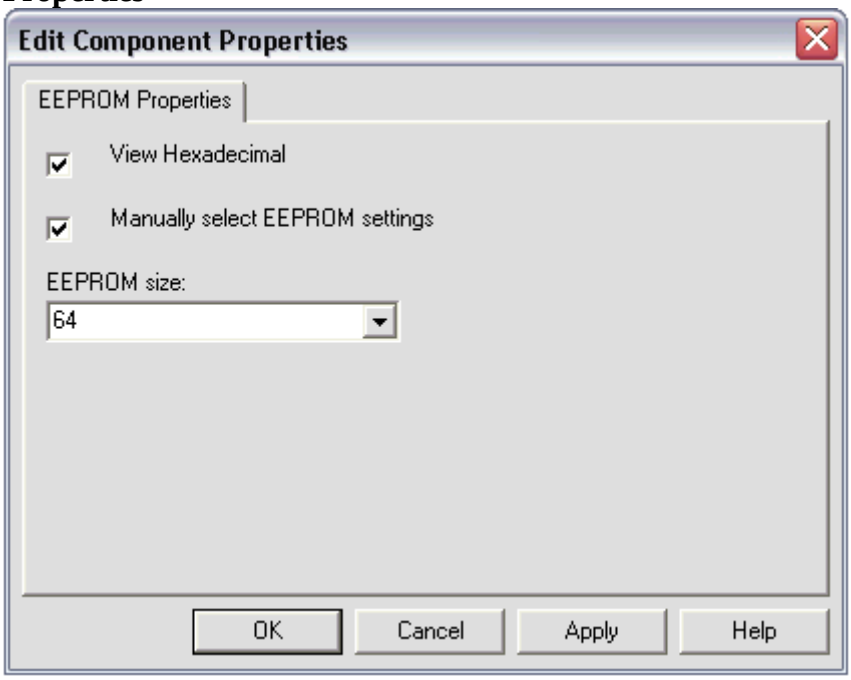
Macros
The Add Defines component has the following macros:

ReadEEPROM(addr)
Reads the data value from the specified address location (addr).

EEPROMWrite(addr, data)
Writes the data value (data) to the specified address location (addr).

Important:
The EEPROM component does not recognize the target PIC straight away. Once you have added the EEPROM component please simulate the program briefly as this refreshes the target PIC setting for the component.

Properties



The Add Defines has the following properties:

View Hexadecimal
The addresses and data are displayed in Hexadecimal format (00 - FF).

Manually select EEPROM settings
Allows the user to set the EEPROM size manually.

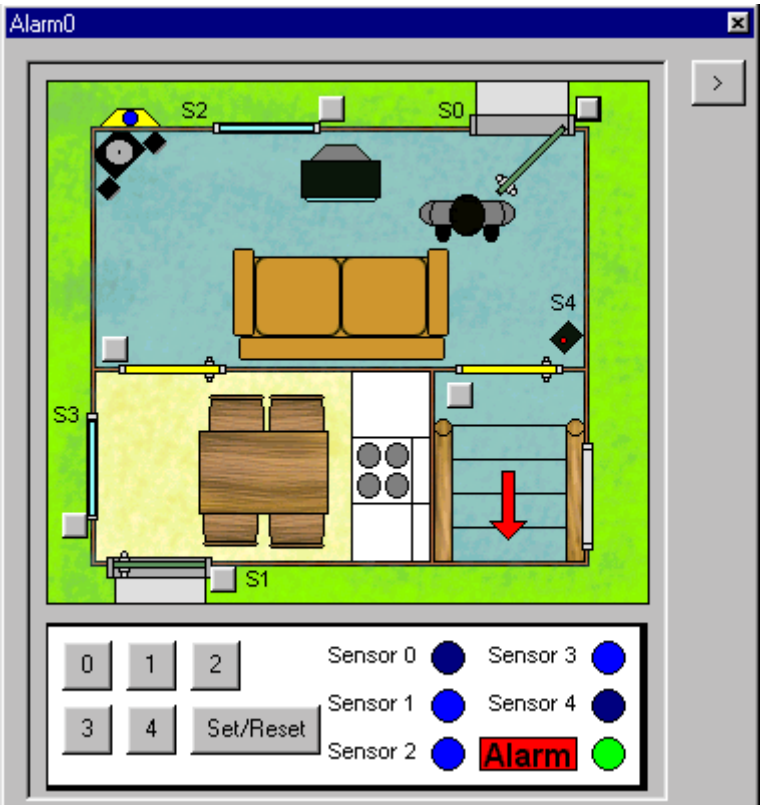
EEPROM size
Allows the user to set the EEPROM size.

Note: These last two properties are intended for use with newer devices that are not on the list of devices supported, but which have EEPROM.
You will need to check that the Device has EEPROM, and how much EEPROM is has.

List of supported devices
The following is a list of PICmicro's with EEPROM that the component can auto-configure itself for:
12ce673, 12ce674, 12f629, 12f675, 12f635, 12f683, 16ce623, 16ce624, 16ce625, 16cr83, 16cr84, 16f627, 16f627a, 16f628, 16f628a, 16f648a, 16f630, 16f676, 16f636, 16f639, 16f684, 16f688, 16f83, 16f84, 16f84a, 16f87, 16f88, 16f818, 16f819, 16f870, 16f871, 16f872, 16f873, 16f873a, 16f874, 16f874a, 16f876, 16f876a, 16f877, 16f877a

Alarm

The component simulates a burglar alarm system fitted to the ground floor of a house.



The component consists of a keypad and LED display to allow input and output, and a Virtual House in which you can open and close windows and doors to simulate the actions of an intruder.

Inputs and outputs
There are 5 sensors marked S0 to S4.

S0 to S3 : Each of these is simply a microswitch on a door or window: when the door or window is closed then the switch is closed. When a door is opened then the switch is also opened. When a door is closed then the LED is turned on – this means that the LED's on the panel can be used to see which sensor has been triggered.
The sensor switches can be simulated by opening/Closing the corresponding door or window in the Virtual House.

S4 : S4 represents a passive infrared detector. For the sake of convenience we shall assume that the detector gives out a logic high (+5V) under normal operation (no burglar) and a logic low (0V) when movement is detected. This means that for the whole alarm when a LED goes off it indicates which sensor her been triggered.
In the Virtual House the IR sensor will respond to any movement in the front room, including opening the stairs and kitchen doors.

Key pad
Keys 0 – 5 and SET / CLEAR are connected to Port B inputs RB0 to RB6 in a standard configuration. This simple keypad gives 99999 different combination codes which should be enough for our purposes. The 100k resistors pull each input pin down to 0V and each push-to-make switch puts 5V on the appropriate line.

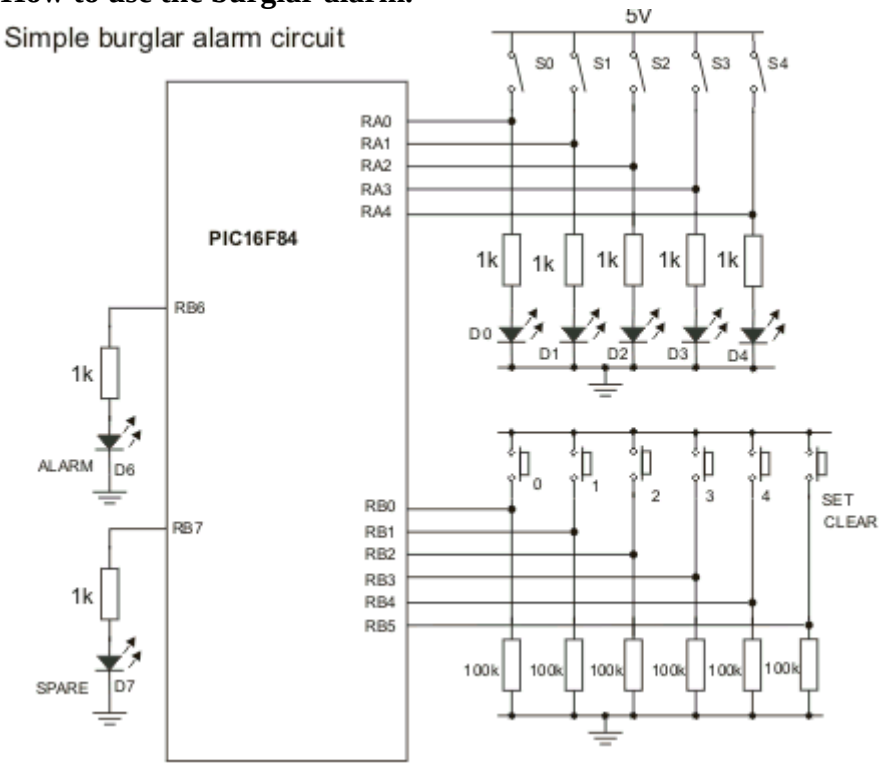
Alarm
RB 6 is used as the alarm output to simply light an LED.
We have used a back lit ALARM panel on the component, but you could use an LED.

SpareLED
RB7 is a spare indicator which can be used for a function of your choice.
On the component the SpareLED is the green LED next to the ALARM light.

How to use the burglar alarm.
The application provides you with a basic alarm circuit. There is no 'correct' way to program this circuit – you can chose any algorithm you like. However we suggest you implement some or all of the following features:

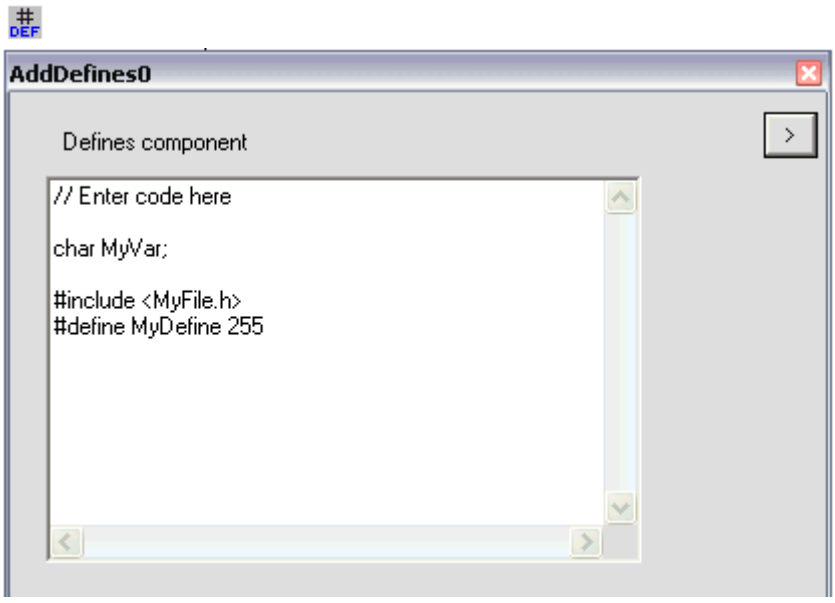
- Hold SET / CLEAR down for more than 10 seconds and then allow the user to enter a sequence of 5 numbers representing the alarm key code.
- Activate the alarm by putting in a sequence of 5 numbers followed by SET / CLEAR.
- Deactivate the alarm, once triggered, by putting in a sequence of 5 numbers followed by SET / CLEAR
- Build a 24 hour clock and automatically activate and deactivate the clock at certain times of day.
- Flash the output at RB7 when the alarm is active.
- Build in a hidden key code that, when entered, tells your program to ignore one sensor.
- Build in a code that sets the alarm for only one rooms (an alarm zone)
- Build in a 30 second period during which the alarm is arming, allowing you to leave the building.

How to use the burglar alarm.



The circuit assumes a PIC16F84 chip is being used, adjust accordingly if another PICmicro microcontroller is being used.

Add Defines



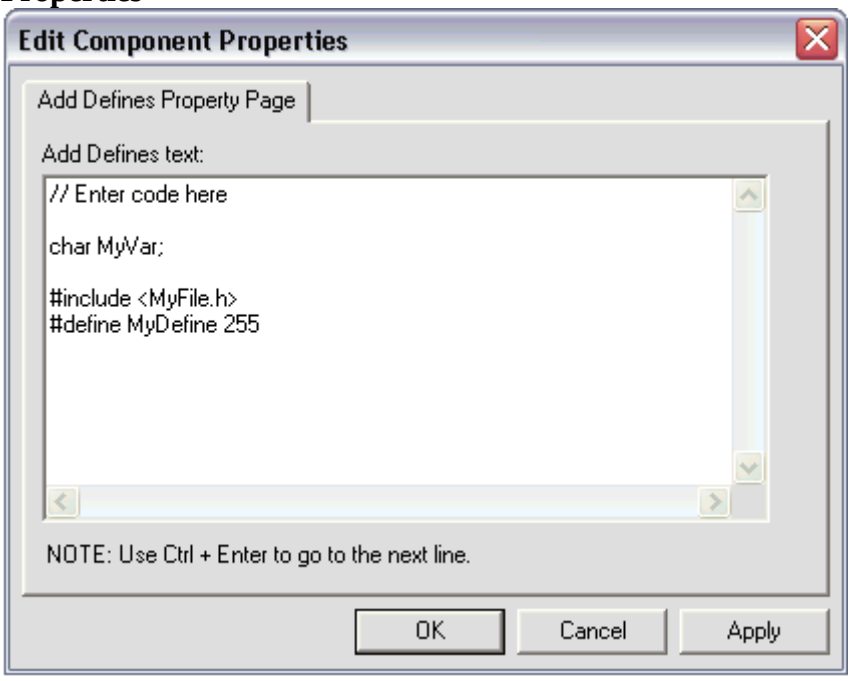
The Add Defines component enables users to add defines, global variables and other such items into the program.
These can then be accessed within 'C' code' icons throughout the user's program.
The component shows the code that will be added to the program.
To edit the code select the '>' button and then "properties' from the menu.

Macros
The Add Defines component has the following macros: AddDefines()
Adds the defines code to the program.

Important: This macro needs to be called within your Flowcode program, otherwise the defines code will not be added to the program.
It does not matter where this function is called.
The macro does not take any parameters, and returns no values.

Note: Neither this macro, nor the component can be simulated.

Properties



The Add Defines has the following properties:

Add Defines text
The code to be added to the program.

Important: The code is not validated or checked in any way by the component. The user is responsible for entering correct code. If you receive compile errors, or your program behaves erratically you may need to check the code that you have entered.

Note: To go to the next line whilst editing the code press Ctrl + Enter.
Enter alone will move the focus out of the text box rather than creating a new line.

Keypad

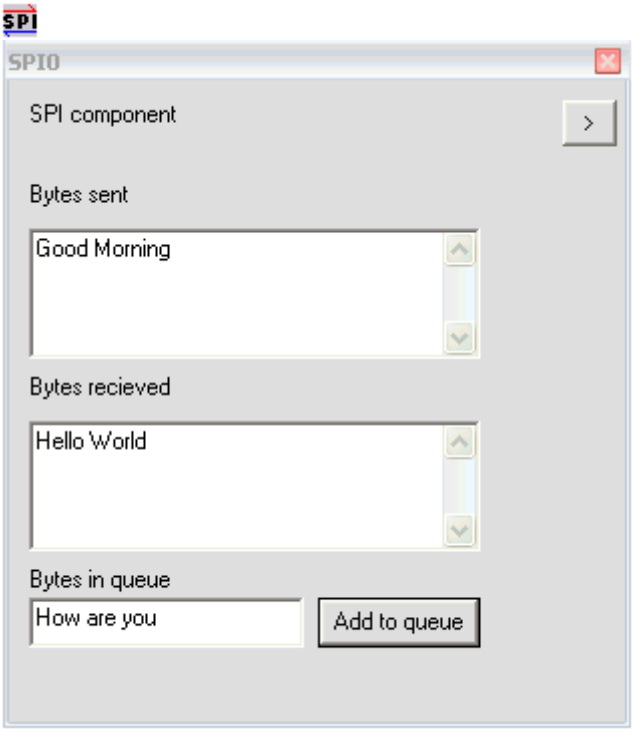
The Keypad is a 3 line 4 button matrix. To determine if a specific key has been pressed you can output a signal onto either the row that the key icon and check if there is a signal on the corresponding column. If there is then the button has been pressed making the connection.

Macros
The Keypad Component provides the following macros:

GetKeypadNumber() : Retrieves the number of the key that has been pressed. The number is 0-9 for the digits, 10 for * and 11 for #

GetKeypadAscii() : Retrieves the ascii value for the character of the key pressed

SPI



The SPI component allows the use of the SPI communications protocol in their programs.
The SPI component was designed to complement the Matrix E-blocks SPI board EB-013.
For more information on setting up and using the SPI E-block please refer to the documentation that accompanied the SPI E-block.
Or visit the Matrix Multimedia website at www.matrixmultimedia.com for the latest documentation and updates.

To help with using the E-blocks SPI board extra macros have been added to facilitate use of the NVM (Non-Volatile Memory, often called FRAM) and DAC (Digital to Analogue Conversion) features of the SPI board.

Note: Not all PICmicro devices have SPI capabilities. Please see the datasheet of the device in question to see if it has SPI capabilities.

Macros
The Add Defines component has the following macros:

SPI_Init() : Initialises the SPI component.

SPI_Uninit() : Uninitialises the SPI component.

SPI_TxByte(byte) : Transmits the SPI data (byte).

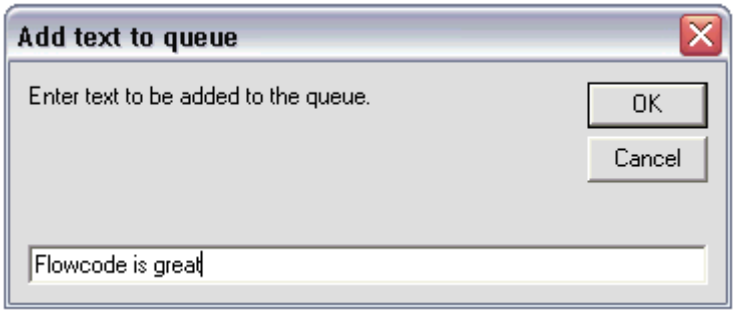
char SPI_RxByte()
Reads the data sent and places the value into the specified return variable.

DAC_TxByte(byte)
Sends the data value (byte) to the DAC output.

NVM_TxByte(hi_addr, lo_addr, data)
Sends the value (data) to the NVM memory location specified by the high address byte (hi_addr) and the low address byte (lo_addr)

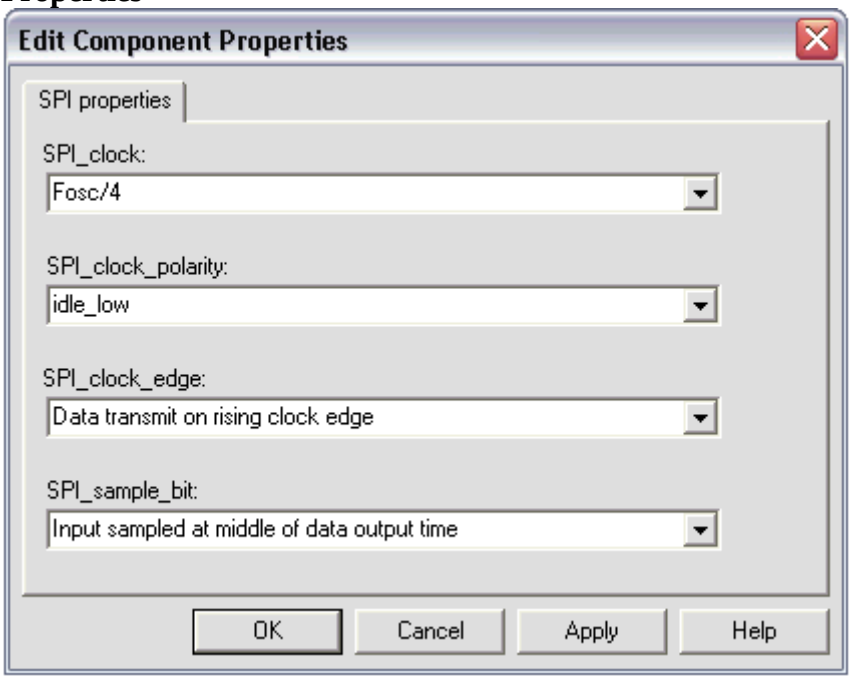
char NVM_RxByte(hi_addr, lo_addr)
Receives the data from the NVM memory location specified by the high address byte (hi_addr) and the low address byte (lo_addr).
The data byte is placed in the return value specified.

Simulating the SPI component
The SPI component simulates SPI communications as follows:
The component displays any text sent from your program in the "Bytes received" box.
The component sends bytes from the text that is in the "Bytes in queue" box to your program or 255 if no data is present.
Bytes sent are also put into the "Bytes sent" box. this allows you to see what has been sent, as well as what is ready to send.



Enter the text into the box and press OK to add the text to the queue.

Properties



The Add Defines has the following properties:

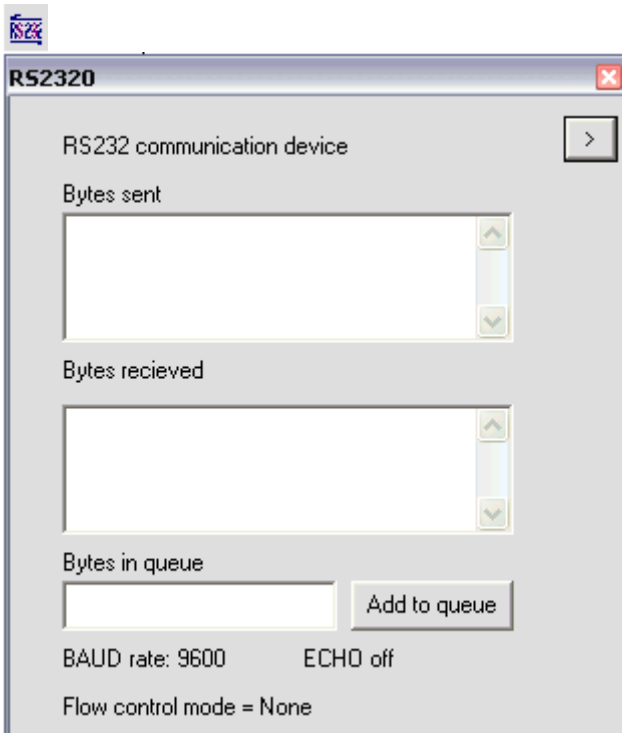
SPI_clock, SPI_clock_polarity, SPI_clock_edge and SPI_sample_bit

These properties are intended for advanced users, and are best left as is unless you are familiar with SPI communications.

Further information on SPI

Please refer to the device datasheets, the SPI board datasheet and Microchips tutorials on SPI.

RS232



The RS232 component is a communications device used to communicate between the PICmicro and another RS232 equipped device.

This could be another PICmicro with an RS232 connection, a PC, or any other device that can communicate via RS232.

Note that RS232 hardware needs to be connected to the PICmicro development board for the RS232 code to function when downloaded to the PICmicro.

This component was designed to function in conjunction with the Matrix RS232 E-block EB-015.

For more information on setting up and using the RS232 E-block please refer to the documentation that accompanied the RS232 E-block or visit the Matrix Multimedia website at www.matrixmultimedia.com for the latest documentation and updates.

Macros

The RS232 Component provides the following macros:

SendRS232Byte(byte) : Sends the byte value via the RS232 connection

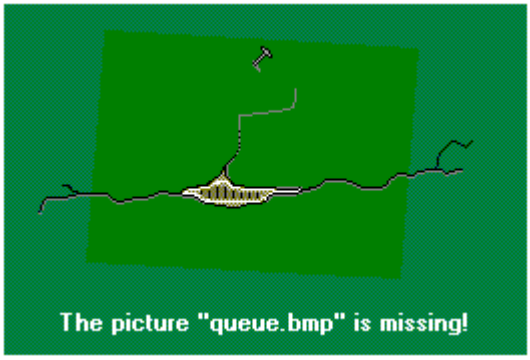
char ReceiveRS232Byte(timeout)

Receives the next byte of data from the RS232 connection.
Timeout specifies how long to wait for a byte of data.
The macro returns the data value or 255 if there was no data received.

Simulating the RS232 component

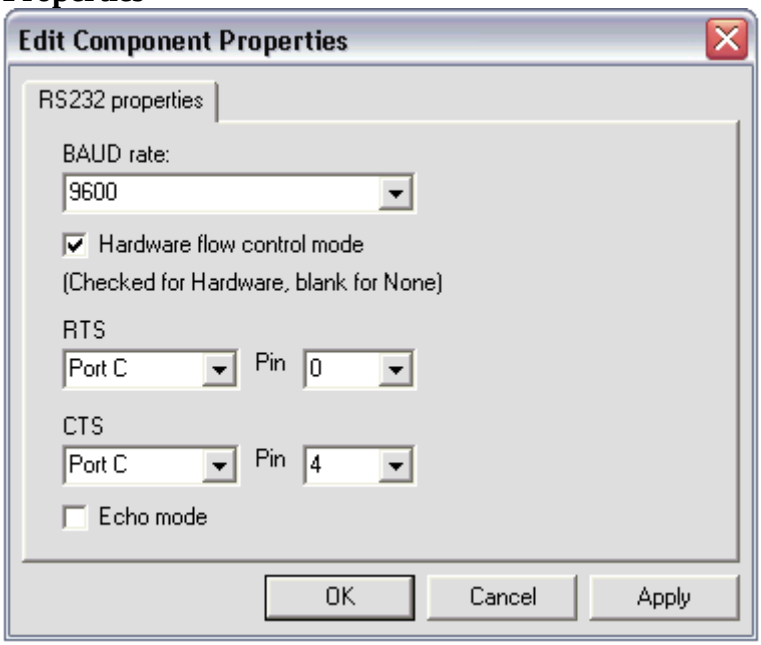
The RS232 component simulates RS232 communications as follows:
The component displays any text sent from your program in the "Bytes received" box.
The component sends bytes from the text that is in the "Bytes in queue" box to your program or 255 if no data is present.

Bytes sent are also put into the "Bytes sent" box. this allows you to see what has been sent, as well as what is ready to send.



Enter the text into the box and press OK to add the text to the queue.

Properties



The RS232 component has the following properties:

Baud rate : Sets the communications rate for the RS232 component.
Note: for accurate communications both the sending and receiving units should be operating at the same baud rate.

Hardware flow control : Enabled Hardware flow control.
For information on flow control settings please consult the documentation on the relevant PICmicro device.

CTS and RTS settings
Allows users to set with Port and which Pin to use for CTS and RTS connections. Note the CTS and RTS settings are disabled and are not shown on the component when Hardware control is not selected.
For information on the CTS and RTS settings please consult the documentation on the relevant PICmicro device.

Echo mode : Enables ECHO mode.

VI) Spécifier le PICmicro cible à programmer

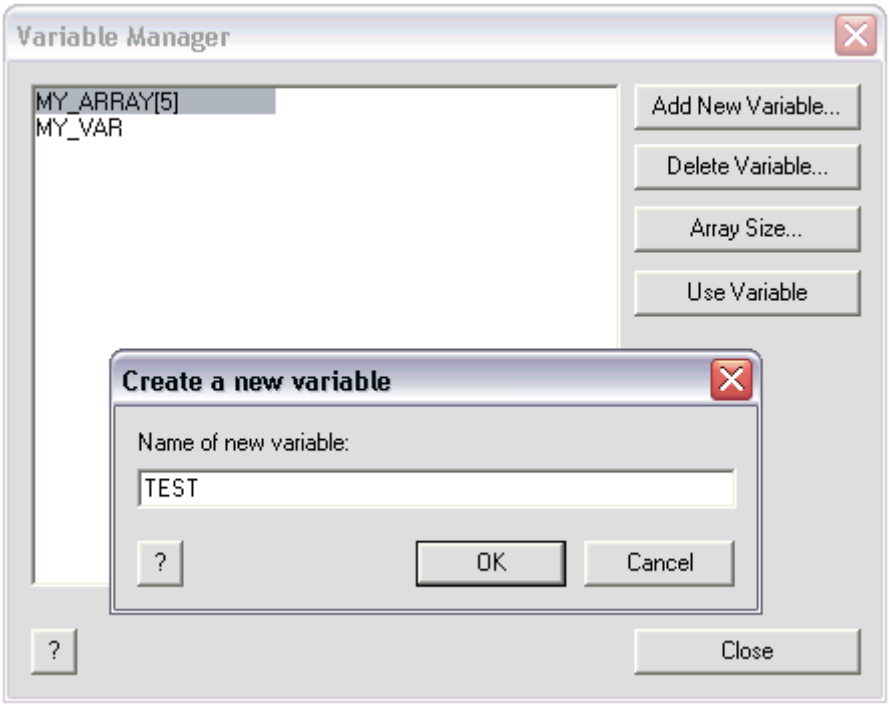
VI-1) Spécifier le PICmicro cible

Utiliser cette option pour spécifier quel microcontrôleur sera programmé par cet organigramme. Si vous changez de microcontrôleur en cours de route, Flowcode vérifie que toutes les actions de votre organigramme sont supportées par le nouveau microcontrôleur.

Flowcode vous avertira s'il détecte un problème potentiel. D'une façon similaire, Flowcode vérifiera que tous les composants externes connectés au microcontrôleur initial pourront être connectés aux même pattes du nouveau microcontrôleur choisi.
Si certaines pattes ne peuvent être connectées, Flowcode vous conseillera de modifier la connexion avant de changer de microcontrôleur.

VII) Travailler avec des macros et des variables

VII-1) Créer des variables



Sélectionner la commande 'Variables' du menu 'Edition' pour faire apparaître la liste des variables déjà définies dans votre organigramme. De nouvelles variables peuvent être définies en utilisant le bouton Ajouter une Nouvelle Variable. Le nom des variables peut comporter 32 caractères alphanumériques au maximum. Le nom peut contenir le caractère souligné (_) mais aucun espace. Il doit aussi comporter au moins une lettre pour pouvoir le distinguer d'un nombre.

Les variables peuvent être supprimées à partir du moment où elles ne sont plus utilisées dans aucune icône de l'organigramme. Flowcode vous délivrera un message d'avertissement si vous essayez de supprimer une variable utilisée dans l'organigramme.

Tableaux
Les tableaux peuvent être créés en ajoutant la taille du tableau entre crochets après le nom de la variable, par exemple MON_TAB[10] crée un tableau de nom MON_TAB composé de 10 éléments.

Vous pouvez manipuler chaque membre du tableau dans l'icône de Calculs en utilisant le nom du tableau suivit du numéro de l'élément entre crochets, par exemple MON_TAB[3] = 32 Notez que les tableaux commencent à 0, c'est-à-dire que tableau de 10 éléments utilisera les nombres 0 à 9 pour numéroter les éléments.

Vous pouvez modifier la taille du tableau en cliquant d'abord sur la variable en question, puis sur le bouton "Taille tableau". Dans cette situation, vous pouvez uniquement modifier la taille et pas renommer la variable.

VII-2) Créer une nouvelle macro

Choisir la commande 'Nouvelle' du menu 'Macro' ou appuyer sur Ctrl+M pour créer une nouvelle macro. Flowcode vous demandera un nom pour cette nouvelle macro, puis ouvrira une nouvelle fenêtre pour vous permettre d'ajouter des icônes à cette macro.

Si vous entrez un nom qui existe déjà pour cette nouvelle macro, Flowcode vous demandera d'entrer un nom différent.

VII-3) Editer et supprimer les macros

Editez ou supprimez des macros existantes en choisissant la commande 'Editer/Supprimer' du menu 'Macro'. Flowcode affichera la liste des macros. Sélectionner le nom d'une macro dans la liste puis cliquer sur le bouton Editer ou le bouton Supprimer.

Lorsque vous éditez une macro, Flowcode affichera la macro dans une nouvelle fenêtre pour vous permettre de la modifier.

Si vous cliquez sur le bouton Supprimer, Flowcode vérifiera que la macro n'est appelée dans aucune autre icône de votre organigramme et délivrera un message d'erreur si c'est le cas.

VII-4) Exporter une macro

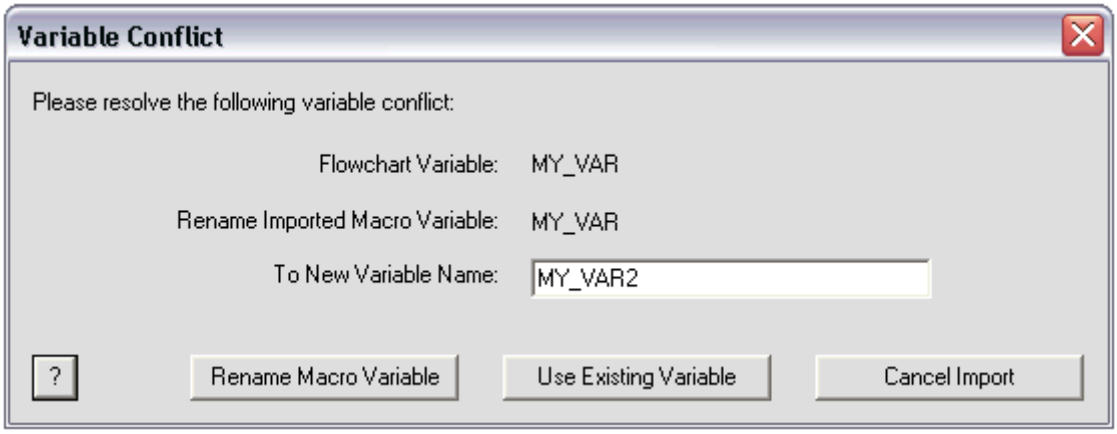
Vos macros (celles que vous avez créées) peuvent être exportées dans un fichier pour un usage ultérieur. Sélectionner la commande 'Exporter' du menu 'Macro'. Flowcode vous demandera d'abord de sélectionner la macro que vous voulez exporter et vous demandera ensuite un nom de fichier. Les macros Flowcode sont exportées dans des fichiers portant l'extension .FCM

VII-5) Importer une macro

Pour importer dans Flowcode une macro précédemment exportée, sélectionner la commande 'Importer' du menu 'Macro'. Flowcode vous demandera de choisir le nom du fichier de la macro existante. Flowcode importera alors la macro et lui attribuera le même nom que la macro exportée. Si le nom de la macro existe déjà, alors Flowcode vous demandera de spécifier un nouveau nom.

Toutes les variables utilisées dans la macro seront ajoutées au programme principal, ce qui vous permettra de les utiliser dans le reste du programme.

Si le nom d'une variable existe déjà, dans ce cas un message d'avertissement apparaîtra et il vous sera demandé de confirmer l'import. Si vous confirmez l'import, il vous sera demandé si vous souhaitez renommer ces variables.

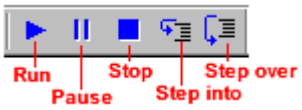


Renommer le nom d'une variable de macro change le nom en celui contenu dans le champ texte.

Utiliser les variables existantes considère que les variables importées depuis la macro référencent les variables existantes de même nom.

VIII) Simuler un algorigramme

VIII-1) Démarrer la simulation de l'algorigramme



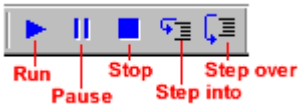
Pour simuler un algorigramme, sélectionnez la commande 'Exécuter/Continuer' du menu 'Exécuter' ou appuyer sur **F5**. Flowcode passe en mode simulation et l'exécution des icônes de l'algorigramme. Un rectangle rouge encadre l'icône en cours d'exécution. Les fenêtres des variables, de la pile des appels et du PICmicro sont mises à jour à chaque étape de la simulation. De plus, si des composants externes sont connectés au microcontrôleur alors leur état est montré dans la fenêtre des composants externes.

Si vous avez demandé à Flowcode de simuler votre application aussi vite que possible alors les fenêtres des variables, de la pile des appels et les vues du microcontrôleur ne sont pas rafraîchies à moins que vous ne suspendiez le déroulement de la simulation.

Par ailleurs, si vous souhaitez simuler votre algorigramme pas à pas depuis le début, alors utilisez la commande 'Pas à pas détaillé' du menu 'Exécuter' ou appuyez sur **F8**.

Voir Changer la vitesse de simulation pour modifier la vitesse de simulation.

VIII-2) Icônes de simulation pas à pas

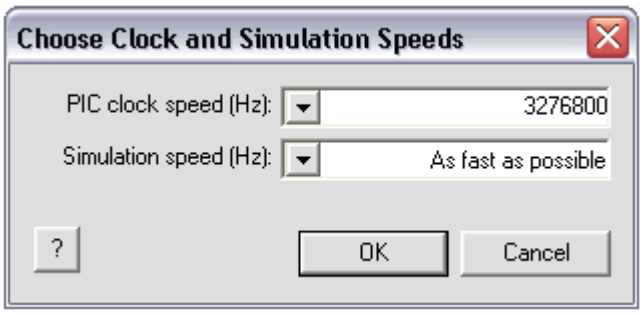


Pas à Pas détaillé
Pour simuler l'application de façon détaillée, icône par icône, sélectionnez la commande 'Pas à Pas détaillé' du menu 'Exécuter' ou appuyer sur la touche F8. Un rectangle rouge signale l'icône qui va s'exécuter. Les variables, la pile d'appel et les fenêtres PIC sont mises à jour à chaque pas. De plus, si des composants externes sont connectés au microcontrôleur, alors leur état sera montré dans la fenêtre des composants externes.

Pas à Pas principal
Pour exécuter une macro dans sa totalité, utiliser la commande 'Pas à Pas principal' du menu 'Exécuter' ou appuyer sur les touches **Maj + F8**. L'option Pas à Pas principal fonctionne un peu comme Pas à Pas détaillé, à la différence suivante: quand le mode Pas à Pas principal rencontre une macro contenant un algorigramme séparé, Pas à Pas principal traite la macro en entier au lieu de l'ouvrir et de l'exécuter pas à pas comme le ferait Pas à Pas détaillé.

Pas à Pas Principal est pratique pour éviter d'avoir à faire un Pas à Pas détaillé dans une macro très longue et permet à l'utilisateur de se concentrer sur d'autres parties de l'algorigramme qui demandent plus d'attention.

VIII-3) Modifier la vitesse de simulation

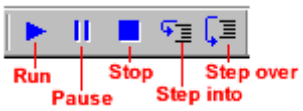


Vitesse d'horloge en Hz
Choisir la vitesse d'horloge depuis le menu PIC pour changer la vitesse de la simulation. La vitesse de simulation peut varier de 0.25 Hz (soit une simulation toutes les 4 secondes) à 10Hz (soit une simulation toutes les 0.1 secondes).

Vitesse de Simulation
De plus, vous pouvez demander à Flowcode de simuler votre diagramme aussi vite que possible. Dans ce mode, Flowcode ne rafraîchit plus les fenêtres des variables, de la pile d'appel et les vues du microcontrôleur.

Information de Configuration
Cliquer sur le bouton Configurer dans la boite de dialogue Vitesse d'horloge pour accéder à l'écran de configuration du programme de téléchargement du PICmicro.
Un certain nombre d'options sont disponibles pour vous permettre de modifier les paramètres de téléchargement. Pour plus de détails sur ces options, reportez-vous à la section Configurer le PICmicro.

VIII-4) Suspendre et arrêter la simulation



Les simulations peuvent être suspendues ou arrêtées grâce aux commandes correspondantes du menu 'Exécuter'.

Dans le cas où l'algorigramme est simulé 'aussi vite que possible', Flowcode mettra à jour les variables, la pile des appels et les vues du microcontrôleur lorsque la simulation est suspendue. Un rectangle rouge apparaîtra autour de la prochaine icône à exécuter.

VIII-5) Ajouter et utiliser des points d'arrêts

Ajouter et supprimer des points d'arrêts (breakpoints)
Les points d'arrêts peuvent être placés ou ôtés en sélectionnant l'icône sur laquelle vous voulez vous arrêter, puis en choisissant l'option Mettre/enlever un point d'arrêt du menu Edition. Une autre façon de faire est d'utiliser la touche fonction F9.

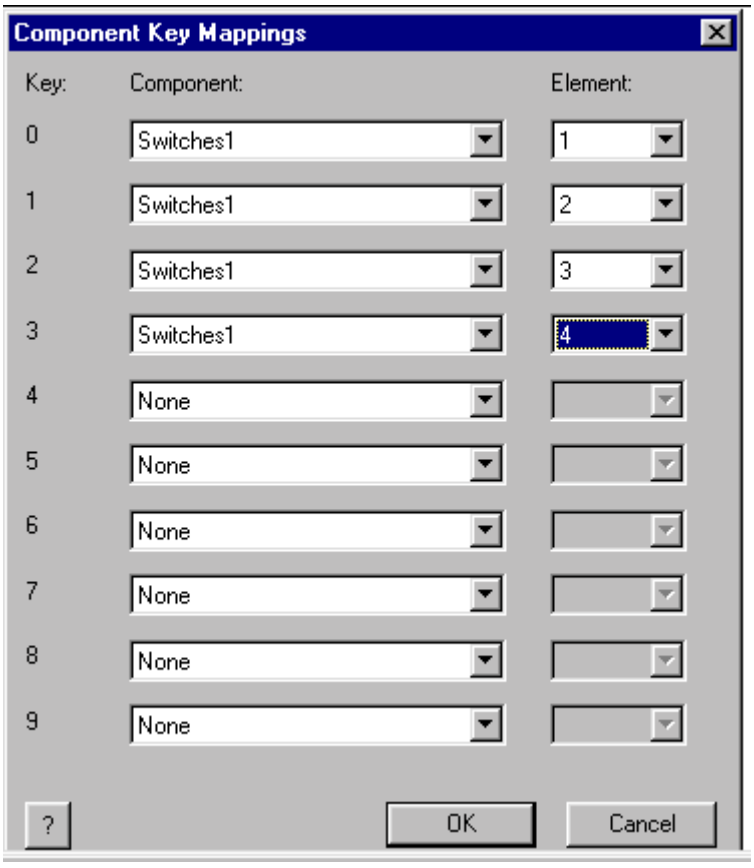
Pour effacer tous les points d'arrêt, sélectionner la commande 'Effacer tous les points d'arrêts' du menu 'Edition'.

Utiliser les points d'arrêts (breakpoints)
Lorsque qu'un algorigramme est simulé, la simulation s'arrête sur le premier point d'arrêt rencontré. Appuyer sur Exécuter/continuer du menu Exécuter pour poursuivre le déroulement de la simulation jusqu'à rencontrer le prochain point d'arrêt s'il existe, ou jusqu'à la fin dans le cas contraire.

Lorsque la simulation rencontre un point d'arrêt, l'utilisateur peut examiner la valeur des variables ou des entrées/sorties, etc.

Les points d'arrêts sont bien pratiques pour arrêter le programme au début d'un morceau de code qui pose problème. L'utilisateur peut alors prendre la main pour avancer pas à pas et mettre au point son code

VIII-6) Editer les raccourcis clavier



La commande 'Touche 0 à 9' du menu 'Edition' vous permet d'utiliser les touches numériques de 0 à 9 pour piloter des composants attachés au microcontrôleur. Par exemple, si vous disposez d'une rangée de 8 boutons poussoirs connectés au microcontrôleur, vous pouvez choisir d'utiliser les touches 1 à 8 pour simuler l'enfoncement puis le relâchement de ces Interrupteurs lorsque l'algorigramme est simulé.

Pour définir ou modifier ces raccourcis clavier, sélectionner la commande 'Touches 0 à 9' du menu 'Edition'. Pour chaque touche numérique, vous pouvez sélectionner un composant et

pour chaque composant vous pouvez sélectionner l'élément que vous souhaitez contrôler par cette touche. Par exemple, si vous voulez utiliser la touche '0' pour contrôler le cinquième Interrupteur du composant nommé 'Interrupteurs1' alors choisissez 'Interrupteurs1' dans la liste des composants pour la touche 0 et ensuite sélectionner le cinquième élément !

Touche
La touche du clavier que vous souhaitez définir.

Composant
Le composant attaché pour lequel vous voulez définir une touche clavier.

Elément
L'élément lui-même (Interrupteurs etc.) auquel vous voulez faire correspondre une touche numérique.

Les éléments sont numérotés séquentiellement dans le même ordre que sur le composant.
Les composants munis de barre de progression (comme le thermomètre) utilisent les éléments 1 et 2 pour augmenter/diminuer la progression

Utiliser une touche clavier avec un composant comme les interrupteurs à bascule permet de passer l'état de On à Off.
Utiliser une touche clavier avec un composant comme les interrupteurs à poussoirs mettra l'élément au niveau haut tant que la touche la touché sera appuyée. En la relâchant la touche, le niveau redeviendra bas donc dans un état inactif.

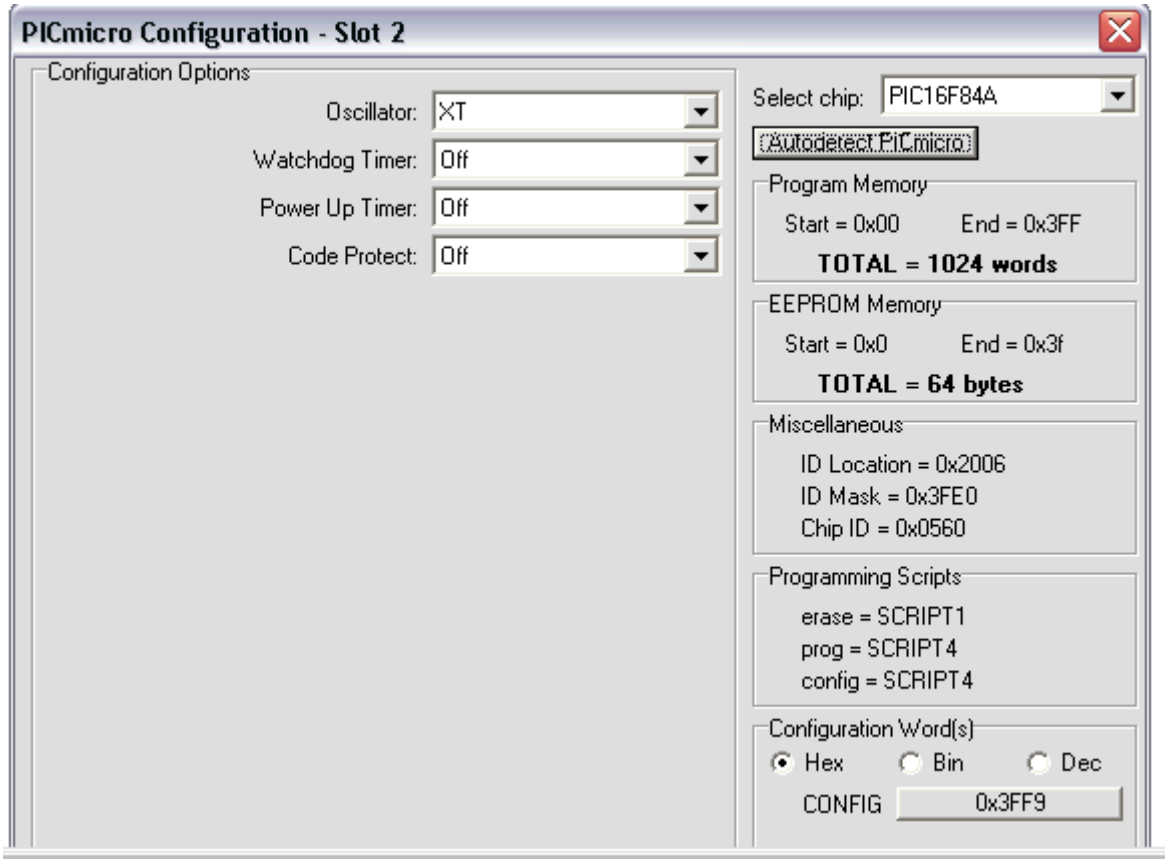
IX) Compiler un algorigramme vers une cible PICmicro

IX-1) Configurer le PICmicro

Le PICmicro peut être configuré via la commande 'Configurer' du menu 'PIC';

Veiller à toujours vérifier la configuration d'un PIC avant d'y télécharger un programme. Vérifier toujours que le type de PICmicro ainsi que le type d'oscillateur (oscillator type) conviennent à votre programme puisque souvent c'est la principale source des soucis rencontrés par les utilisateurs.

Configurer le PICmicro avec le programme PPP de téléchargement par défaut
La commande 'Configurer' du menu 'PIC' affiche la page de configuration.



Cette page permet de:

Spécifier le composant PIC à utiliser
Définir le type d'oscillateur
Valider/Invalider les timers
Définir la version de la carte de développement utilisée (boutons options en bas)

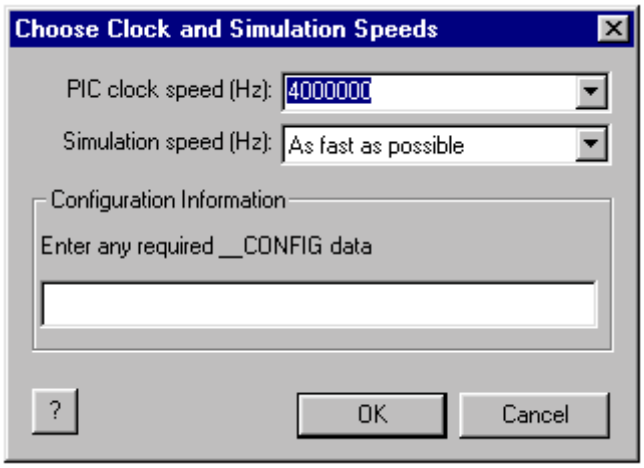
Si vous utilisez Flowcode pour la première fois, ou avez changé de carte de développement, vous devriez contrôler les options relatives à la carte utilisée.

D'autre options sont également disponibles (voir le fichier d'aide pour plus de détails).

Vous trouverez des informations supplémentaires dans le fichier d'aide de PPP. Vous y avez accès en cliquant sur le bouton d'aide '?' dans le coin en bas et à gauche de la page de configuration.

Configurer le PICmicro avec d'autres programmes de téléchargement

Si vous n'utilisez pas le programme de configuration PPP alors l'écran suivant s'affiche.



Vous pouvez spécifier les informations de configuration dans la boite de dialogue fournie.

Vous devrez consulter la documentation de votre programme de téléchargement pour plus de détails sur les options et les commandes à utiliser.

IX-2) Compiler un algorigramme vers un PICmicro

Rubrique vide, contacter le revendeur pour la mise à jour du fichier d'aide.

IX-3) Spécifier les options de compilation

Si vous utilisez la carte de développement Matrix, vous n'aurez pas besoin d'utiliser cet écran puisque tous les réglages sont faits au moment de l'installation.

Si vous utilisez d'autres systèmes de développement alors vous pouvez paramétrer Flowcode pour créer un lien direct vers le logiciel approprié. Les informations suivantes vous seront alors bien utiles :

Compilateur C
Spécifie l'emplacement de votre compilateur C. Cette définition est normalement faite de façon automatique lors de l'installation de Flowcode. Vous ne devriez pas avoir à le modifier.

Fichiers d'Include
Spécifie l'emplacement des fichiers d'include dont le compilateur C a besoin. Ce paramètre est normalement défini de façon automatique lors de l'installation de Flowcode. Vous ne devriez pas avoir à le modifier.

Assembleur
Spécifie l'emplacement du programme d'assemblage – MPASM. Ce paramètre est normalement défini de façon automatique lors de l'installation. Le suffixe permet de transmettre des informations à MPASM donnant des détails sur les options de programmation et le type de composant cible.

Programmateur
Spécifie l'emplacement du programme permettant de télécharger le code hexadécimal de votre programme dans le PICmicro cible.

Si vous utilisez MPLAB® avec PICSTART® alors dans la ligne concernant l'assembleur vous devrez entrer l'emplacement du programme exécutable MPLAB. Le répertoire par défaut est :
C:\Program Files\MPLAB IDE\MCHIP_Tools\MPLAB.EXE

Ceci fait que si vous sélectionnez la commande 'Compiler vers PIC' du menu 'Exécuter', le fichier ASM sera traité dans l'environnement MPLAB. A partir de là, vous devrez assembler votre programme (en utilisant MPASM) et le télécharger dans le PICmicro avec PICSTART.

Par contre, si vous spécifiez comme emplacement:
C:\Program Files\MPLAB IDE\MCHIP_Tools\MPASMWIN.EXE
alors votre fichier .ASM sera automatiquement assemblé.

Les options de compilation par défaut sont listées ci-dessous et peuvent être réintroduites dans Flowcode à tout moment en cliquant sur le bouton « Restaurer défauts ».

Compilateur:
C:\Program Files\Matrix\Flowcode\c2c\c2c.exe -PPIC%p -IP%p.inc -ndi
Include:
C:\Program Files\MPLAB IDE\MCHIP_Tools\P%p.inc
Assembleur:
C:\Program Files\MPLAB IDE\MCHIP_Tools\Mpasmwin.exe /aINHX8M /p%p /rHEX /w2 /q

Programmateur:

C:\Program Files\Matrix\PPPv3\pppv3.exe -cs 2

Les suffixes présents dans ces lignes de commandes permettent de rendre le processus de compilation et de téléchargement des programmes vers les PICmicro aussi transparents que possibles pour les utilisateurs de la carte de développement de Matrix Multimedia.

Problème répertorié : Compiler avec des droits utilisateur restreints
Si vous utilisez Flowcode avec des droits utilisateur limités ou restreints vous devrez contacter votre administrateur système pour obtenir des privilèges d'accès en lecture/écriture à la clé de registre HKEY_LOCAL_MACHINE\Software\Licenses.
Ce problème concerne les utilisateurs de Windows 2000 et Windows XP.

X) Tutoriaux

X-1) Utiliser les tutoriaux

Flowcode est livré avec plus de 20 tutoriaux qui vous aideront dans le développement de vos propres applications pour PICmicro. Nous vous suggérons de passer du temps à étudier ces tutoriaux dans l'ordre numérique de façon à bien assimiler le fonctionnement de Flowcode. Une liste des tutoriaux ainsi que leurs fonctions est donnée dans la section Liste des tutoriaux.

Par défaut, les travaux dirigés sont conçus pour un microcontrôleur PIC16F84.

Bien que les programmes puissent tourner sur la plupart des autres microcontrôleurs PICmicro, nous ne garantissons pas qu'ils fonctionnent sur tous les composants de la gamme.

Le PIC16F84 a été choisi parce qu'il s'agit d'un composant relativement simple et complet.

Les tutoriaux ont été conçus dans l'esprit d'utiliser la carte de développement V2 de Matrix Multimedia. La carte est équipée des périphériques d'entrée/sortie classiquement utilisés par les programmeurs: un interrupteur et une LED pour chaque bit du Port A et du Port B, un quadruple afficheur 7 segments, un afficheur LCD et des entrées de conversion analogique – numérique pour capteurs. Cette carte de développement peut être utilisée à la fois pour programmer des PICmicros différents et pour développer et tester vos applications.

Tous les tutoriaux ont été testés avec la carte de développement V2 de Matrix Multimédia.

Les tutoriaux mettant en oeuvre des entrées analogiques nécessitent un microcontrôleur capable de les gérer. Dans ce cas, c'est le microcontrôleur PIC16F877 qui a été retenu.

Le mieux est de simuler les algorigrammes en pas à pas, puisqu'ainsi vous percevrez mieux le fonctionnement du microcontrôleur et la façon dont les variables sont affectées.

Faire fonctionner les applications des tutoriaux sur la carte de développement vous amènera peut être à ajouter des pauses (delay) afin de ralentir la vitesse d'horloge.

Dans tous les cas, si vous apportez des modifications à l'un des tutoriaux, il est préférable de sauvegarder le fichier modifié sous un nom différent afin de ne pas modifier le fichier original.

X-2) Liste des tutoriaux

TUT_01.FCF Tutorial 1: Allumer une LED
- Allumer une LED (A0). Démontre comment envoyer une sortie à un port.
- Effacer le PORT A en envoyant 0, ce qui éteint toutes les LEDs connectées au PORT A.
- Puis envoyer 1 au PORT A, ce qui allumera la LED A0

TUT_02.FCF Tutorial 2: Envoyer une valeur sur un port.
- Envoyer 5 sur le PORT A, ce qui allumera deux LEDs.
- Les deux LEDs allumées correspondent à la combinaison binaire 5 (soit 101).

Essayez ceci :
- *Changer la valeur dans l'icône SORTIE et observer le résultat sur la rangée de LEDs*

TUT_03.FCF Tutorial 3: Bits individuels et ports.
- Envoyer 1 à chaque bit du PORT A, puis effacer le port A. Finalement, allumer toutes les LEDs reliées au PORTA.
- Envoyer 1 sur un bit particulier, ce qui allumera la LED connectée à ce bit.

Essayez ceci :
- *Changer l'ordre dans lequel les LEDS sont allumées.*
- *Allumer d'autres LEDs.*

TUT_04.FCF Tutorial 4: Utiliser des variables.
- Envoi de MY_OUTUPUT au PORT A.
- Envoyer une variable vers un port est équivalent à envoyer une valeur.

Essayez ceci :
- *Changer la valeur affectée à MY_OUTPUT dans l'icône CALCUL.*
- *Ajoutez une nouvelle variable OUTPUT_A et changer le calcul pour utiliser cette variable.*

TUT_05.FCF Tutorial 5: Calculs élémentaires.
- Exécute des opérations de base et envoie le résultat sur le PORT B.

Essayez ceci:
- *Modifier le calcul ou en ajouter de nouveaux.*
- *Ajouter une deuxième variable et l'utiliser dans des opérations impliquant plus d'une variable.*

TUT_06.FCF Tutorial 6: Entrées interrupteurs.
- Lire une entrée reliée à un interrupteur connecté au PORT A et l'envoyer au PORT B.
- Notez que ce tutorial utilise aussi une boucle pour tourner indéfiniment.

Essayez ceci :
- *Essayer de ne lire qu'un seul bit du port plutôt que le port entier.*

TUT_07.FCF Tutorial 7: Calculs de logique booléenne.
- Déterminer RESULT à partir d'une variable, d'une entrée du PORT A et d'un ET logique.

Essayez ceci :
- *Modifier la valeur de BOOLVAR.*
- *Essayez d'autres fonctions booléennes comme OR et XOR.*

TUT_08.FCF Tutorial 8: Utiliser des masques.
- Transmet les entrées lues sur le PORT A au PORT B, mais applique un masque pour ne sélectionner que certains bits du PORT A.

Essayez ceci :
- *Modifier les bits masqués.*
- *Voyez ce qui arrive lorsque vous appliquez un masque sur la sortie.*

TUT_09.FCF Tutorial 9: Un compteur élémentaire.
- Incrémente un compteur et visualise le résultat sur les LEDs connectées au PORT B.
- Notez que lorsque notre compteur atteint 255 (toutes les LEDs sont allumées), il déborde et repart à 0.

TUT_10.FCF Tutorial 10: Un compteur temporisé.
- Identique au tutorial 9, mais avec une pause de 1 seconde pour ralentir le comptage

Essayez ceci :
- *Modifier la valeur de la pause pour accélérer ou ralentir le comptage.*

TUT_11.FCF Tutorial 11: Utiliser des boucles.
- Ce compteur s'appuie sur une boucle pour compter jusqu'à 16 et afficher le résultat sur le PORT A.

Essayez ceci :
- *Connecter les LED's au PORT B et augmenter la valeur du compteur.*

TUT_12.FCF Tutorial 12: Un chenillard élémentaire.
- La lumière se propage de LED en LED.
- Le nombre binaire suivant est calculé en multipliant le précédent par 2. Ce calcul nous permet d'allumer successivement les LEDs.

Essayez ceci :
- *Pouvez-vous le faire fonctionner à l'envers ?*

TUT_13.FCF Tutorial 13: Un chenillard amélioré.
- Dans le tutorial précédent, la lumière s'arrêtait à la fin de la rangée de LED. Nous avons donc ajouté une icône décision pour faire repartir le compteur.

Essayez ceci :
- *Comment faire pour faire la même chose mais à l'envers ?*

TUT_14.FCF Tutorial 14: Décalage de bit.
- Un autre chenillard, mais les valeurs sont calculées en utilisant les opérateurs de décalage de bits.
- En fait, nous avons utilisé les opérateurs pour passer d'un bit au suivant.

Essayez ceci :
- *Modifier la valeur de comptage pour voir ce qui se passe.*
- *Modifier le sens de décalage des bits.*

TUT_15.FCF Tutorial 15: Chenillard avec des boucles (1).
- Presque identique au précédent, mais intégrant des icônes de Boucles.

Essayez ceci :
- *Comparer ce tutorial avec le suivant pour voir différentes façons de résoudre le même problème.*

TUT_16.FCF Tutorial 16: Chenillard avec des boucles (2).
- Toujours notre même chenillard, mais élaboré à partir des icônes Décision et Points de jonction.

Essayez ceci :
- *Comparer ce tutorial avec le précédent pour voir différentes façons de résoudre le même problème.*

TUT_17.FCF Tutorial 17: Utiliser un afficheur 7 segments.
- Afficher un nombre sur un afficheur 7 segments.
- Nous utilisons une macro qui nous permet d'afficher directement un nombre donné sur l'afficheur.

Essayez ceci :
- *Jeter un coup d'œil aux propriétés de la macro.*
- *Modifier le digit affiché.*
- *Afficher le point décimal.*

TUT_18.FCF Tutorial 18: Compter sur un afficheur 7 segments.
- Utilise une simple boucle pour utiliser l'afficheur 7 segment comme compteur.

TUT_19.FCF Tutorial 19: Comptage qui utilise les interruptions.
- Dans ce tutorial, nous utilisons une interruption basée sur le timer pour afficher un chiffre chaque seconde.
- La vitesse d'horloge est réglée sur 3276800Hz, et la valeur du pré-diviseur est de 1:128
- Ces réglages délivrent une interruption à la fréquence de 25Hz (25 fois par seconde)
- Nous pouvons alors éditer macro de l'interruption TMR0 pour mettre à jour la variable affichée toutes les secondes.

Essayez ceci :
- Changer la Vitesse d'horloge et la valeur du pré-diviseur pour voir la répercussion sur la fréquence d'interruption du timer.
- *Changer le code pour qu'il fonctionne avec ces nouvelles valeurs.*

TUT_20.FCF Tutorial 20: Comptage sur un quadruple afficheur 7 segments avec multiplexage.
- Le PICmicro ne peut piloter qu'un seul des quatre digits à la fois. Cependant, le PICmicro peut piloter successivement chacun des 4 afficheurs afin qu'ils semblent continuellement allumés pour l'œil humain. Cette technique s'appelle le multiplexage.
- Notez que cet exercice utilise la macro UPDATE_VALUES.
- Un des objectifs des macros est de conserver un algorigramme lisible en plaçant des parties de code trop importantes ou trop complexes dans un autre algorigramme spécifique. Cette technique présente aussi l'avantage de pouvoir réutiliser des parties de code.

Essayez ceci :
- *Modifier le code de façon à supprimer le multiplexage. A quoi ressemble votre afficheur maintenant ?*
- *Créer deux nouvelles macros - une pour initialiser l'afficheur et une pour afficher.*

TUT_21.FCF Tutorial 21: Utiliser l'afficheur LCD.
- Utilise l'afficheur LCD pour afficher un message.

Essayez ceci :
- *Modifier le message (voir la page d'aide de l'afficheur LCD pour plus de détails).*

TUT_22.FCF Tutorial 22: Une horloge 24 heures.
- Affichage de l'heure sur l'afficheur LCD (format 24 heures).
- Notez comment les macros améliorent l'apparence et simplifient la lecture de votre algorigramme.

Essayez ceci :
- *Modifier le code pour que votre horloge fonctionne en 12 Heures AM / PM (matin / après-midi).*

TUT_23.FCF Tutorial 23: Interruption PORT B0.
- Utilise une interruption sur B0 du PORT B pour incrémenter un compteur.

Essayez ceci :
- *Modifier le code de telle façon que le compteur soit continuellement mis à jour et qu'il soit remis à 0 par une interruption.*

TUT_24.FCF Tutorial 24: Générer un son.
Note: Flowcode ne prend pas en charge la simulation des sons.
- Transforme les interrupteurs du PORT B en touche de piano.
- Le son est généré en excitant la patte A0.
- Connecter un écouteur ou un haut-parleur à la sortie audio pour que le son soit audible.

Essayez ceci :
- *Modifier les valeurs de TONE et écoutez l'effet sur le son.*

TUT_25.FCF Tutorial 25: Code C et assembleur enfouis.
- Cet exercice explique l'utilisation du langage C et du code assembleur enfouis.

Note: Le C et le code ASSEMBLEUR ne sont pas pris en charge par la simulation avec Flowcode.

TUT_26.FCF Tutorial 26: Utiliser les entrées analogiques.
Note: Cet exercice nécessite un PICmicro équipé d'une entrée analogique, comme le PIC16F877.
- Lit le niveau de l'entrée analogique et l'affiche sur l'afficheur LCD.

TUT_27.FCF Tutorial 27: Calculs avancés.
- Cet exercice montre un certain nombre d'opérations complexes.

TUT_28.FCF Tutorial 28: Utiliser des macros.
- Cet exercice explique l'utilisation des macros.

X-3) Utiliser les tutoriaux avec d'autres PICmicro

Pour modifier le tutorial afin de l'utiliser avec un autre PICmicro, il faut:

1) Sélectionner le nouveau composant PIC - voir Spécifier un PIC cible.

2) Valider la vitesse d'horloge du composant PIC - voir Modifier la vitesse de simulation.

3) Configurer le programme de téléchargement pour le nouveau composant PIC – voir Configurer le PICmicro.

4) Contrôler tous les composants associés afin de s'assurer qu'ils sont connectés correctement.

Si le nouveau composant PIC ne possède pas les mêmes ports, ou ne contient pas les mêmes fonctionnalités (telle une entrée analogique) , les composants ne seront pas connectés correctement ou deviendront indisponibles.

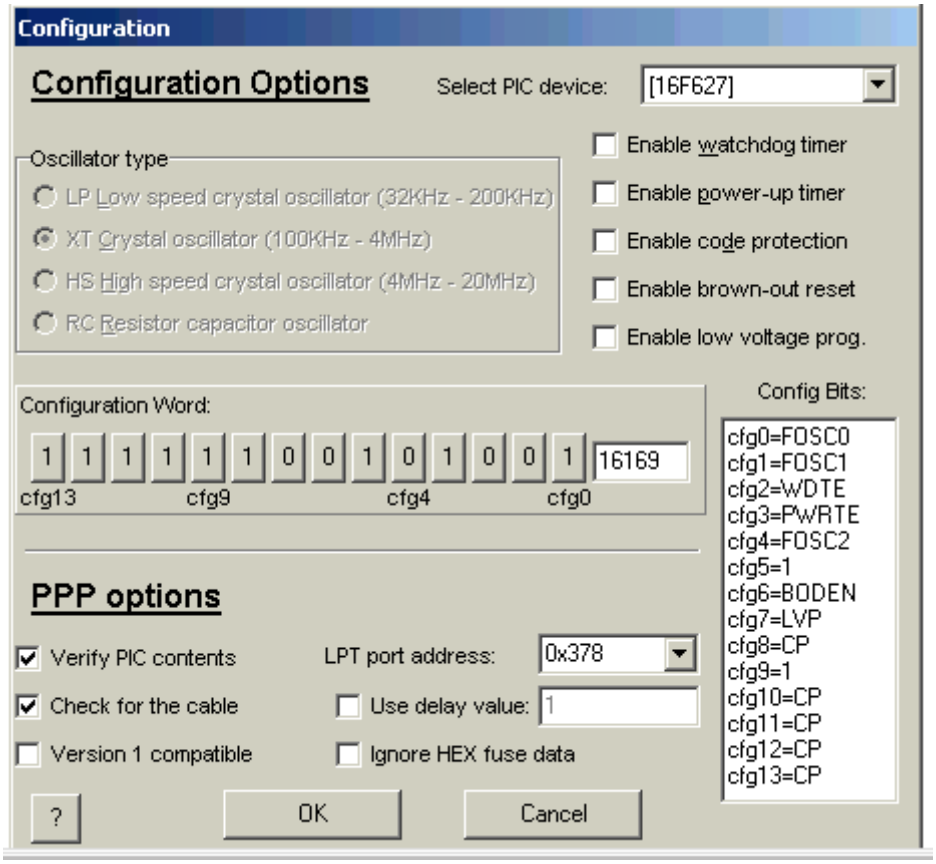
X-4) Utiliser les tutoriaux avec les composants PIC16F627 et PIC16F628

Les composants PIC16F627 et PIC16F628 nécessitent des réglages particuliers pour être utilisés avec la carte de développement PICmicro de Matrix Multimedia. Procédez de la façon suivante pour modifier un tutorial de telle sorte qu'il tourne avec un PICmicro de type PIC16F627 ou PIC16F628 :

1) Sélectionner le nouveau PIC à programmer - voir Spécifier le PIC cible.

2) Régler la vitesse d'horloge pour ce PICmicro - voir Changer la vitesse de simulation.

3) Configurer le programmateur de la carte de développement pour les composants PIC16F627 ou PIC16F628.
Ouvrir l'écran de configuration et vérifier que le paramétrage correspond à l'image ci-dessous. Accordez une attention toute particulière au mot de configuration.



4) Vérifier que tous les composants attachés sont correctement connectés.
Si le nouveau PICmicro ne dispose pas des mêmes ports ou ne possèdent pas les spécificités requises, les composants risquent d'être déconnectés ou de devenir indisponibles.

Fichier PIC16F627 de test
Un fichier de test des PIC16F627/PIC16F628 est contenu dans le dossier des tutoriaux : 627_TEST.FCF.
Ce fichier de test montre l'utilisation du comparateur AD présent sur la série des PIC16F627.
Ce fichier contient du code en C et par conséquent ne peut être simulé.
Le test nécessite une entrée analogique Entrée ANA0.
Le test utilise le capteur de lumière figurant sur la carte de développement Matrix pour détecter s'il fait jour ou pas. Les LEDs du Port B s'éclairent s'il fait jour. Recouvrez le capteur pour éteindre les LEDs.